

Faisal

It is recommended that this book be distributed to all  
of the Board's Secondary Education Schools

for Grade

10

Model Test Papers

# COMPUTER SCIENCE

JOIN  
FOR  
MORE!!!

Author:

Muhammad Afzal Ali Syed



Included:

- » Multiple Choice Questions (M.C.Qs)
- » Short Answer Questions
- » Detailed Answer Questions
- » Summary of Lessons
- » Solution of Textbook Exercises
- » Weblinks / Websites
- » Model / S.S.C. Examination Paper



Faisal Publishers  
W.D. Bazar, Karachi

# Contents

Unit	Description	Page	Unit	Description	Page
<b>UNIT 01</b>	<b>Problem Solving &amp; Algorithm Designing</b>		<b>UNIT 05</b>	<b>Functions</b>	
▶ Multiple Choice Questions (M.C.Qs)	05		▶ Multiple Choice Questions (M.C.Qs)	87	
▶ Short & Detailed Answer Questions	09		▶ Short & Detailed Answer Questions	90	
▶ Summary	19		▶ Summary	104	
▶ Solution of Textbook Exercise	20		▶ Solution of Textbook Exercise	105	
▶ Lab Activity	21		▶ Lab Activity	106	
<b>UNIT 02</b>	<b>Basics of Programming In C++</b>		<b>UNIT 06</b>	<b>Digital Logic &amp; Design</b>	
▶ Multiple Choice Questions (M.C.Qs)	23		▶ Multiple Choice Questions (M.C.Qs)	108	
▶ Short & Detailed Answer Questions	28		▶ Short & Detailed Answer Questions	113	
▶ Summary	37		▶ Summary	123	
▶ Solution of Textbook Exercise	38		▶ Solution of Textbook Exercise	124	
			▶ Lab Activity	124	
<b>UNIT 03</b>	<b>Input / Output Handling In C++</b>		<b>UNIT 07</b>	<b>Introduction to Scratch</b>	
▶ Multiple Choice Questions (M.C.Qs)	40		▶ Multiple Choice Questions (M.C.Qs)	127	
▶ Short & Detailed Answer Questions	46		▶ Short & Detailed Answer Questions	132	
▶ Summary	59		▶ Summary	140	
▶ Solution of Textbook Exercise	60		▶ Solution of Textbook Exercise	141	
▶ Lab Activity	60		▶ Lab Activity	141	
<b>UNIT 04</b>	<b>Control Structure</b>				
▶ Multiple Choice Questions (M.C.Qs)	63				
▶ Short & Detailed Answer Questions	66				
▶ Summary	82				
▶ Solution of Textbook Exercise	83				
▶ Lab Activity	84				

▶ Model / S.S.C. Annual Examination Paper



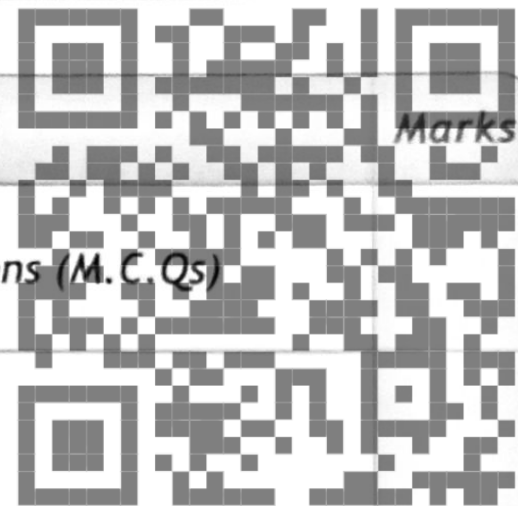


**Board of Secondary Education  
Karachi Sindh.**

**S.S.C. (Annual Examination)  
Secondary Stage Computer Science - 10**

**Division of Marks**

Section	Detail	Marks
A	Multiple Choice Questions (M.C.Qs)	
B	Short Answer Questions	
C	Detailed Answer Questions	
Total Marks		75



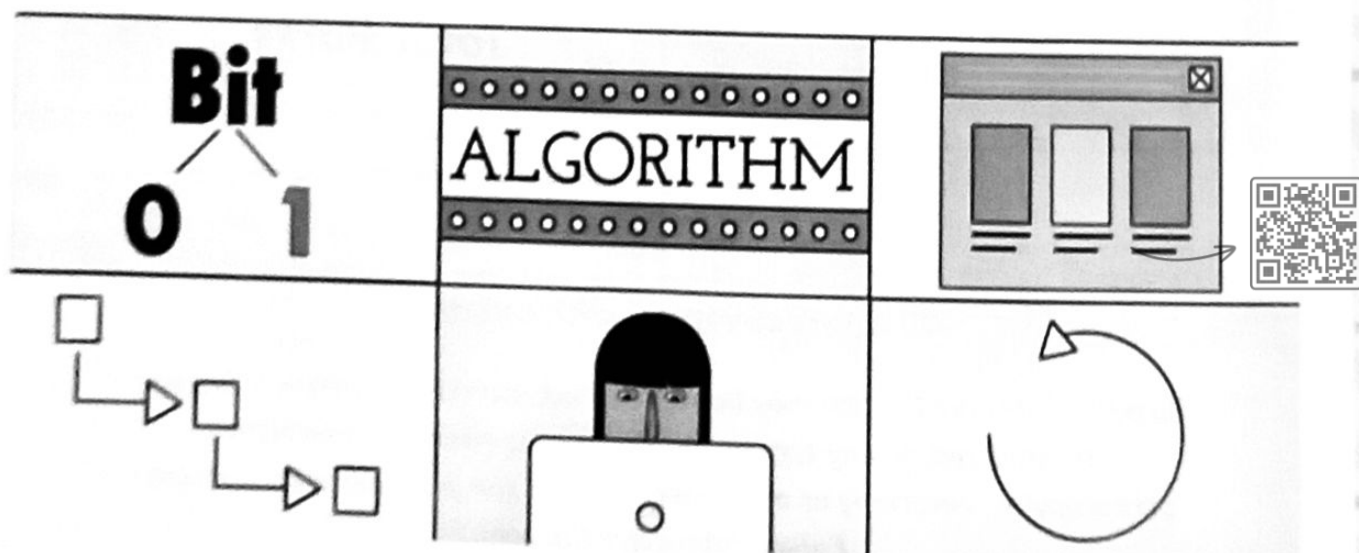
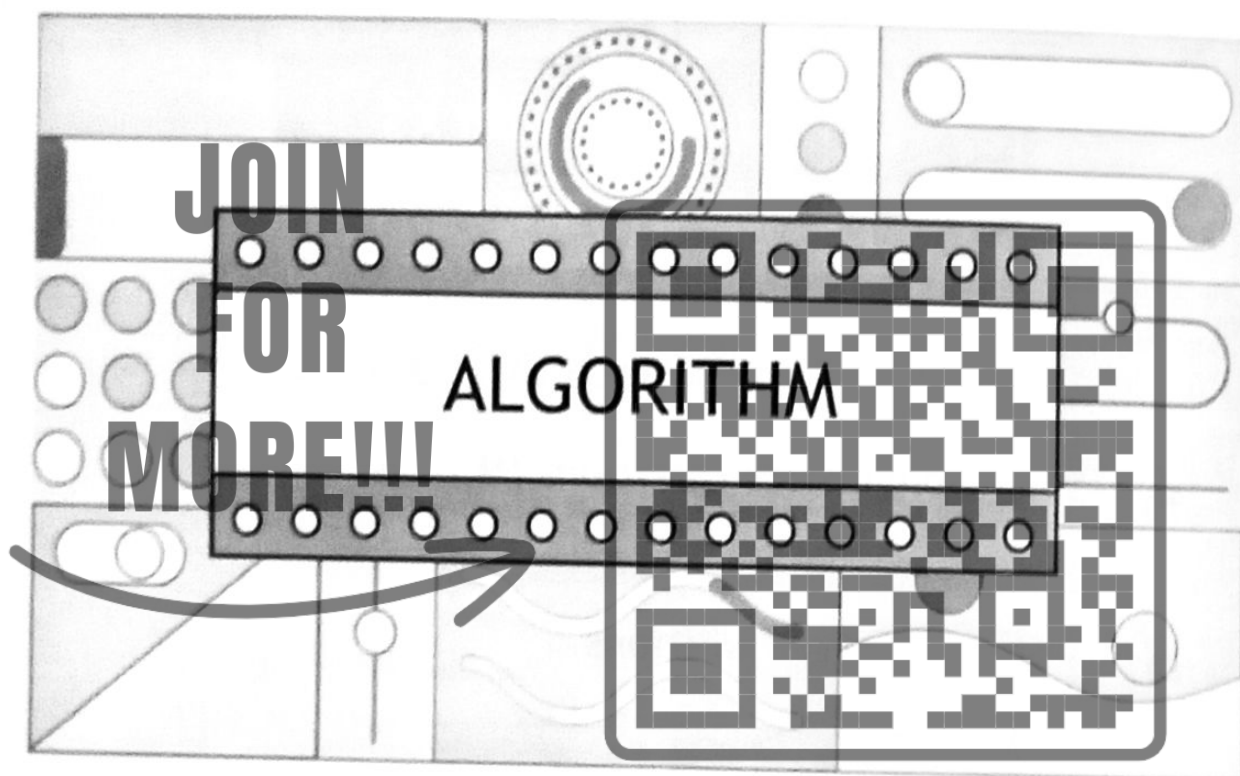
**All rights reserved with publisher**

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means; electronic mechanical, photocopying, recording or otherwise, without the prior permission of the Faisal Publishers Karachi.





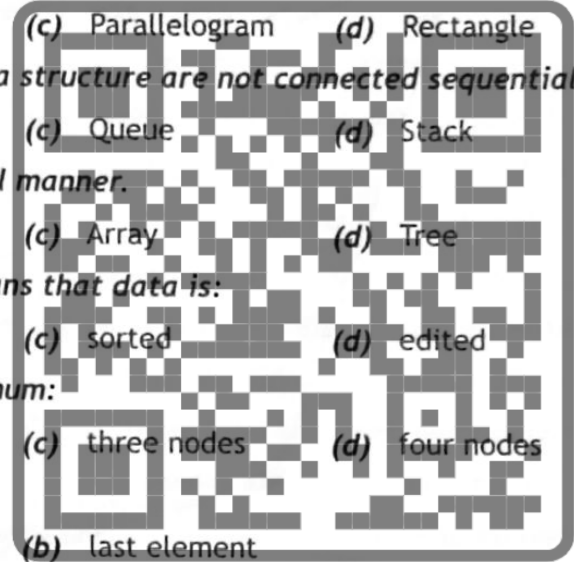
# Problem Solving and Algorithm Designing



# Multiple Choice Questions (M.C.Qs)

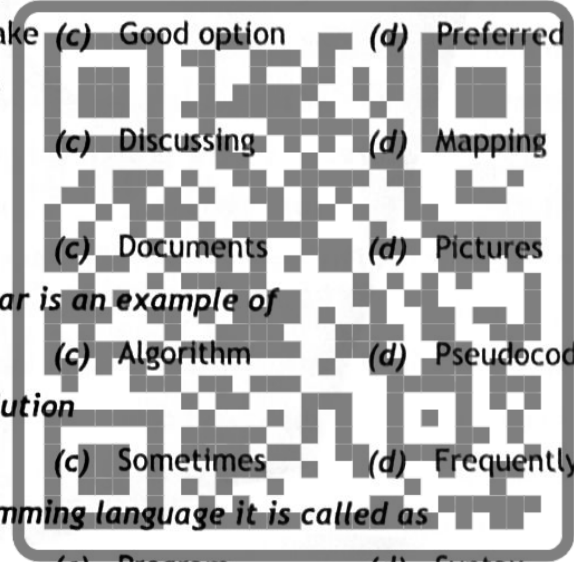
Choose the right answer:

01. To "bake a cake" is an example of:  
(a) problem (b) strategy (c) algorithm (d) solution
02. To find a feasible solution to a problem, the first step is to:  
(a) establish starting point (b) find available solutions  
(c) create a strategy (d) identify and analyze the problem
03. Step by step solution of a problem in simple language is called:  
(a) Problem Solving (b) Algorithm (c) Flowchart (d) Data Structure
04. \_\_\_\_\_ shows the logic of program graphically.  
(a) Data Structure (b) Graph (c) Algorithm (d) Flowchart
05. Symbol is used for input output in flowchart.  
(a) Triangle (b) Square (c) Parallelogram (d) Rectangle
06. In which of the following Elements of data structure are not connected sequentially.  
(a) Array (b) Graph (c) Queue (d) Stack
07. \_\_\_\_\_ stores data in hierarchical manner.  
(a) Stack (b) Queue (c) Array (d) Tree
08. When data is Pushed in stack, it means that data is:  
(a) inserted (b) deleted (c) sorted (d) edited
09. In binary tree, each child can have maximum:  
(a) one node (b) two nodes (c) three nodes (d) four nodes
10. Traversing an array means accessing:  
(a) first element (b) last element  
(c) any specific element (d) each and every element of the array
11. Solving problem is the core feature of  
(a) Mobile (b) Computer (c) Printer (d) Calculator
12. Responsibility of a programmer include  
(a) Setting up project (b) Assign roles  
(c) Analyze Problem (d) Define Problem
13. Problem is defined as a  
(a) Question (b) Resolution (c) Guideline (d) Feasibility
14. A person who plan and strategize task is called as  
(a) System Admin (b) System Analyst (c) Programmer (d) Graphic Designer
15. Which is a very important aspect of problem-solving  
(a) Writing Problem (b) Searching notes  
(c) Developing Strategy (d) Building team

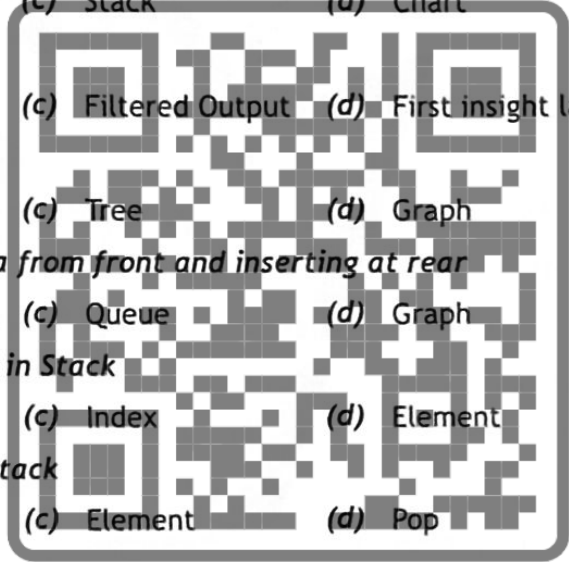




- 16.** How many basic steps are involved in solving a problem  
 (a) One (b) Two (c) Three (d) Four
- 17.** First step in problem solving is termed as  
 (a) Find Alternate (b) Define Problem (c) Select Alternate (d) Implementation
- 18.** Second step in problem solving is termed as  
 (a) Select Alternate (b) Implementation (c) Find Alternate (d) Define Problem
- 19.** Third step in problem solving is termed as  
 (a) Define Problem (b) Find Alternate (c) Implementation (d) Select Alternate
- 20.** Fourth step in problem solving is termed as  
 (a) Implementation (b) Select Alternate (c) Define Problem (d) Find Alternate
- 21.** Defining a problem means identifying  
 (a) Symptoms (b) Real Problem (c) Solution (d) Factors
- 22.** How many possible solutions a problem can have  
 (a) Many (b) Four (c) One (d) Ten
- 23.** Evaluating alternatives as they are proposed is a  
 (a) Best practice (b) Common Mistake (c) Good option (d) Preferred method
- 24.** Follow up on the solution is referred as  
 (a) Tracking (b) Budgeting (c) Discussing (d) Mapping
- 25.** An algorithm is a set of  
 (a) Rules (b) Images (c) Documents (d) Pictures
- 26.** Following traffic rules while driving a car is an example of  
 (a) Flowchart (b) Program (c) Algorithm (d) Pseudocode
- 27.** An algorithm produces best possible solution  
 (a) First time (b) Always (c) Sometimes (d) Frequently
- 28.** When an algorithm is written in programming language it is called as  
 (a) Flowchart (b) Pseudocode (c) Program (d) Syntax
- 29.** Algorithms are expressed in  
 (a) High level language (b) Pseudocode (c) Flowchart (d) Both b&c
- 30.** Flowchart defines the  
 (a) Starting point (b) Sequence (c) Format (d) Hierarchy
- 31.** Pictorial representation of an Algorithm is called as  
 (a) Flowchart (b) Pseudocode (c) Graph (d) Template
- 32.** Which symbol is used to define flow of program in a Flowchart  
 (a) Line (b) Arrow (c) Diamond (d) Dash
- 33.** Oval shaped symbol in a flowchart describes  
 (a) Start (b) Process (c) Input (d) Output
- 34.** Rectangular shaped symbol in a flowchart describes  
 (a) Process (b) Input (c) Output (d) Start



- 35.** Parallelogram shaped symbol in a flowchart describes  
 (a) Start (b) End (c) Process (d) Input
- 36.** Diamond shape in a flowchart is used to write  
 (a) Condition (b) Error (c) Output (d) Counter
- 37.** Which symbol in flowchart is used to show jump from one point to other  
 (a) Diamond (b) Circle (c) Terminator (d) Rectangle
- 38.** Circle in a flowchart is also known as  
 (a) Connector (b) Breake (c) Decision (d) Terminator
- 39.** How many types of data structure are available  
 (a) 1 (b) 2 (c) 3 (d) 4
- 40.** Linear data structure is  
 (a) Random (b) Sequential (c) Preset (d) Multi-level
- 41.** Example of linear data structure is  
 (a) Graph (b) Tree (c) Stack (d) Chart
- 42.** FILO stands for  
 (a) First in least out (b) First in Last out (c) Filtered Output (d) First insight last out
- 43.** LIFO is an example of  
 (a) Queue (b) Stack (c) Tree (d) Graph
- 44.** Which data structure allows deleting data from front and inserting at rear  
 (a) Tree (b) Stack (c) Queue (d) Graph
- 45.** Which term is used to insert new element in Stack  
 (a) Pop (b) Push (c) Index (d) Element
- 46.** Which term is used to delete element in Stack  
 (a) Index (b) Pus (c) Element (d) Pop
- 47.** Queue uses following order of operation  
 (a) LIFO (b) FIFO (c) FILO (d) FOFI
- 48.** Array hold a list of finite data with  
 (a) Different type (b) Same type (c) One type (d) Special type
- 49.** Each item stored in array is called as  
 (a) Pop (b) Index (c) Push (d) Element
- 50.** Numerical value for each array location is called as  
 (a) Index (b) Pop (c) Push (d) Element
- 51.** In order to check data each element of array can be accessed and termed as  
 (a) Search (b) Traverse (c) Deletion (d) Sorting
- 52.** What do you call an array which has all elements in a specific order  
 (a) Sorted Array (b) Traverse Array (c) Simple Array (d) Macro Array
- 53.** Which type of data structure has multiple paths  
 (a) Linear (b) Non-Linear (c) Stack (d) Array



- 54.** FIFO stands for  
 (a) First in Final out (b) First in Filled out (c) First in First out (d) Final in First out
- 55.** How many Queues are required to carry out stack  
 (a) 0 (b) 1 (c) 2 (d) 4
- 56.** Non-linear data structure with hierarchical arrangement between elements is called as  
 (a) Tree (b) Stack (c) Nodes (d) Array
- 57.** Very first node in tree data structure with no further connection is termed as  
 (a) Root (b) Stem (c) Parent (d) Vertices
- 58.** Node in tree data structure with further connection is termed as  
 (a) Child (b) Stem (c) Parent (d) Vertices
- 59.** Node in tree data structure with prior connection is termed as  
 (a) Root (b) Stem (c) Child (d) Vertices
- 60.** Binary tree can have maximum \_\_\_\_\_ children nodes  
 (a) 1 (b) 2 (c) 3 (d) 4
- 61.** In a graph data structure edges connects \_\_\_\_\_ nodes  
 (a) 1 (b) 2 (c) 3 (d) 4
- 62.** There are \_\_\_\_\_ types of graph data structure  
 (a) 1 (b) 2 (c) 3 (d) 4
- 63.** A type of graph data structure in which nodes are connected by edges which are bidirectional is called as  
 (a) Directed (b) Undirected (c) Unified (d) Simple
- 64.** A type of graph data structure in which nodes are connected by edges which are unidirectional is called as  
 (a) Directed (b) Undirected (c) Unified (d) Simple

## Answers

- |         |         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|---------|
| 1. (a)  | 2. (d)  | 3. (b)  | 4. (d)  | 5. (c)  | 6. (b)  | 7. (d)  | 8. (b)  |
| 9. (b)  | 10. (d) | 11. (b) | 12. (c) | 13. (a) | 14. (c) | 15. (a) | 16. (d) |
| 17. (b) | 18. (c) | 19. (d) | 20. (a) | 21. (a) | 22. (a) | 23. (b) | 24. (a) |
| 25. (a) | 26. (c) | 27. (b) | 28. (c) | 29. (d) | 30. (b) | 31. (a) | 32. (b) |
| 33. (a) | 34. (a) | 35. (d) | 36. (c) | 37. (b) | 38. (a) | 39. (b) | 40. (b) |
| 41. (c) | 42. (b) | 43. (b) | 44. (c) | 45. (b) | 46. (d) | 47. (b) | 48. (b) |
| 49. (d) | 50. (a) | 51. (b) | 52. (a) | 53. (b) | 54. (c) | 55. (c) | 56. (a) |
| 57. (a) | 58. (c) | 59. (c) | 60. (b) | 61. (b) | 62. (b) | 63. (b) | 64. (a) |



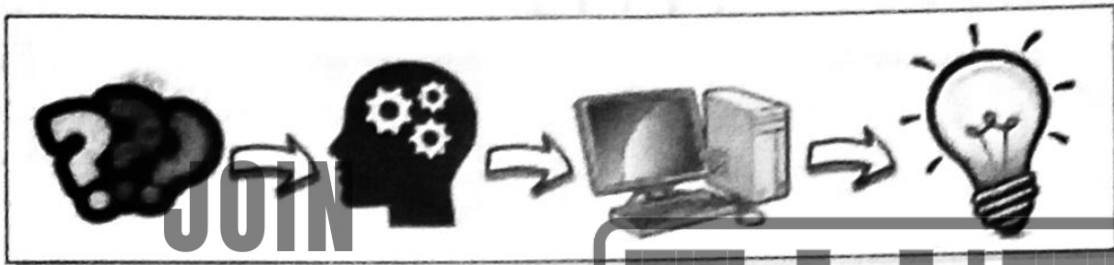
# Short & Detailed Answer Questions

## Q.1 Define the term problem?

**Ans.** A problem is a situation preventing something from being achieved. A problem can be a task, a situation or any other thing. In simple words, a problem is a word which requires answer.

## Q.2 Describe the steps involved in problem solving?

**Ans.** Problem solving is a step-by-step process. There are four basic steps involved in finding a solution for a problem as stated below:



- Define the problem
- Generate alternative solution
- Evaluate and select an alternative solution
- Implement and Follow up on the solution.

### Define the Problem:

In problem solving process the very first step is to define a problem. Identification of a problem is a crucial task and requires proper diagnoses. This step will lead to all other efforts, so a well-described problem is very important.

### Generate alternative solution

A problem can have more than one solution. It is important to postpone the selection of one solution until several problems-solving alternatives have been proposed. More solution options will increase the quality of output. Thinking and team problem-solving techniques are both useful tools at this stage of problem solving.

### Evaluate and select an alternative solution

A common mistake in problem solving is that alternatives are evaluated as they are proposed, so the first acceptable solution is chosen, even if it is not the best solution. If we just focus on the results and ignore evaluating all options, we will miss out on learning new things.

### Implement and Follow up on the solution

Problem solving does not only involve the solution to immediate problem, but it also helps to track, monitor and measure future impact. When choosing most appropriate solution, the problem solver should consider about the possible impacts of that solution. For example, will this solution solve the current problem without creating new ones or if this solution is acceptable by everyone involved in this situation or if the solution is within the budget and achievable within a given time.

### Q.3 Define Algorithm?

**Ans.** An algorithm is a set of instructions/steps/rules that are followed to solve a problem. It is tool for solving a well-specified computational problem.

### Q.4 What are the advantages of developing algorithm?

**Ans.** Algorithms are widely used throughout all areas of information Technology (IT). An algorithm is a set of instructions/steps/rules that are followed to solve a problem. It is a tool for solving a well-specified computational problem. The advantage of using an algorithm to solve a problem or plan so that it produces the best possible answer every time. This is useful in solutions where accuracy is required or where similar problems need to be solved more often. In many cases, computer programs can be developed with the help of this process. Then data is entered in that program so that the algorithm can be executed to come up with the required solution.

### Q.5 State qualities of a good algorithm?

**Ans.** A well written algorithm is essential in order to create a successful code. Following are the qualities of a good algorithm:

- Input and output should be defined precisely.
- Each step in the algorithm should be clear and unambiguous.
- Algorithms are supposed to be most effective among many ways to solve a problem.
- An algorithm should not include computer code. Instead, the algorithm should be written in such a way that it can be used in different programming languages.

### Q.6 Design an algorithm to find the greater number by taking two numbers as input?

**Ans.** **Step 1:** Start  
**Step 2:** Declare variable num1, and num2  
**Step 2:** Read num1, num2  
**Step 3:** If num1>num2 then  
    Display "num1 is the largest number".  
    Otherwise  
    Display "num2 is the largest number".  
**Step 4:** Stop

### Q.7 Design an algorithm to find area of the triangle?

**Ans.** **Step 1:** Start  
**Step 2:** Declare variable side1, side2, side3, sum, and area  
**Step 2:** Read side1, side2 and side3  
**Step 3:** Calculate  $\text{sum} = (\text{side1} + \text{side2} + \text{side3}) / 2$   
**Step 4:** Calculate  $\text{area} = \sqrt{\text{sum} * (\text{sum} - \text{side2}) * (\text{sum} - \text{side2}) * (\text{sum} - \text{side3})}$   
**Step 5:** Stop

**Q.8** Sort the following steps of the algorithm in correct order for baking a cake:

Step: Gather the ingredients

Step: End

Step: Grease a pan

Step: Preheat the oven

Step: Put the pan in the oven

Step: Start

Step: Pour the batter into the pan

Step: When the timer goes off, take the pan out of the oven

Step: Set a timer

Step: Mix together the ingredients to make the batter

**Ans.** Step: Start

Step: Preheat the oven

Step: Gather the ingredients

Step: Mix together the ingredients to make the batter

Step: Grease a pan

Step: Pour the batter into the pan

Step: Put the pan in the oven

Step: Set a timer

Step: When the timer goes off, take the pan out of the oven

Step: End

**Q.9** Define Flowchart?

**Ans.** It is a general-purpose tool used to define the sequence of different types of processes or operations in information system or program. It shows processes and their flow visually using diagram. It describes graphically different steps of programs or any operation and their sequence or flow using different symbols.

**Q.10** What are information system flowchart?

**Ans.** Information system flowcharts show flow of data from source documents to final distribution to users.

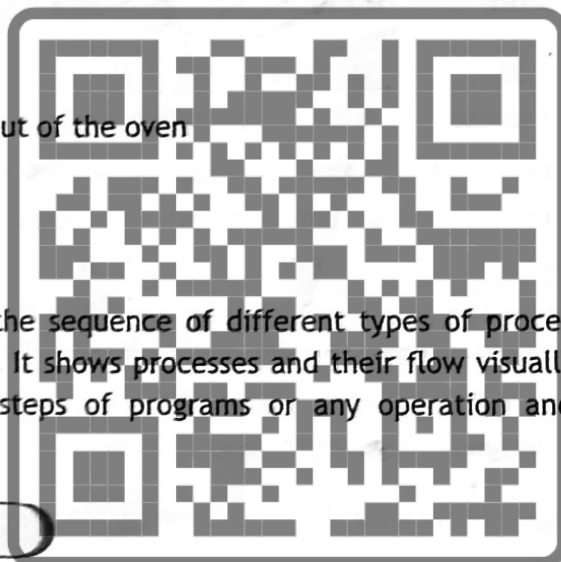
**Q.11** What is a program flowchart?

**Ans.** Program flowcharts show the sequence of steps or instructions in a single program or subroutine. It is diagrammatic or graphical representation of algorithm and converts word of algorithm into symbols.

**Q.12** List any three advantages of creating Flowcharts?

**Ans.** Flowchart is made up of different symbols to represent or show program and its flow. Following are the three major advantages of creating a flowchart:

- Flowcharts help in communicating the logic of a program to all others and make it easy to understand.
- It is a useful program document that is needed for various purposes like to know about program quickly or to modify program logic.





- The flowcharts act as a guide or blueprint during the coding of program.

**Q.13 Describe some basic flowchart symbols?**

**Ans.** Following are some of the basic flowchart symbols:

**Start and end symbol**



This symbol is also referred to as the terminator symbol as it represents starting and end points, as well as potential outcomes of a process path. The start and end symbols will be an elongated oval shape.

**Process symbol**



This common symbol is shaped as a rectangle, and it can also be called the action symbol. It represents an action, function or process and can be considered one of the most-used flowchart symbols.

**Document symbol**



This symbol is shaped like a rectangle with its bottom side in a wave, and it is used to represent the input or output of a document. For instance, this symbol might be used to outline a document input, such as receiving an email or report. Similarly, it can be used to represent a document output like producing a presentation or project.

**Connector symbol**



These symbols are little circles used to connect separate elements of a flowchart across a whole page. Connectors are typically used in flowcharts with more complex processes, such as software or application development.

**Decision symbol**



Shaped as a rhombus, this symbol is used to indicate a question that results in a "yes" or "no" answer, as well as a possible "true" or "false" situation. Depending on the answer to the proposed question, the flowchart can then split into various branches to complete the outline of the workflow.

**Off-page connector or link symbol**



This symbol looks like an upside-down pentagon and is oftentimes used in more complex flowcharts to connect the separate elements of multiple pages. There can be a page number within each shape, allowing for easier reference.

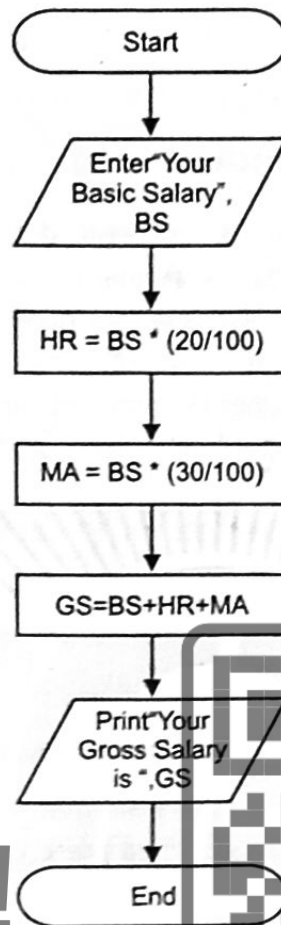
**Input and output symbol**



This symbol is used to represent any data that can be available for input and output. Also referred to as the data symbol, this shape can also be used to represent the resources used or produced. Shaped as a parallelogram, it may sometimes be substituted with the paper tape symbol.

**Q.14** Draw a flow chart to calculate gross salary by adding 20% house rent and 30% medical allowances in basic salary?

**Ans.**



**JOIN  
FOR  
MORE!!!**

**Q.15** What is the difference between Algorithm and Flowchart?

**Ans.** Following table describe the difference between algorithm and flowchart:

Algorithm	Flowchart
Algorithm is step by step solution of a problem	Flowchart is a diagram of different shapes which shows flow of data through processing system.
In algorithm text is used.	In flowchart, symbols or shapes are used.
Algorithm is easy to debug.	Flowchart is difficult to debug.
Algorithm is difficult to write and understand.	Flowchart is easy to construct and understand.
Algorithm does not follow any rules.	Flowchart follows rules for its construction.
Algorithm is the pseudo code of the program.	Flowchart is just graphical or visual representation of program logic.

**Q.16** What is Data structure?

**Ans.** A data structure is a particular way of organizing data in a computer to use it effectively. For example, array data structure is used to store a list of items having the same data-type.

**Q.17** What is linear data structure?

**Ans.** Linear data structure is a data structure type in which elements are arranged in sequential order and each of the elements is connected to its previous and next element.

**Q.18 What is non-linear data structure?**

**Ans.** It is a type of data structure in which elements are not connected in a sequence. Each element can have multiple paths to connect to other elements. They support multi-level storage and often cannot be traversed in single run.

**Q.19 Describe Tree and Graph data structure?**

**Ans.** Tree: Tree data structure is used to represent data containing a hierarchical relationship between elements. Tree represents its elements as the nodes connected to each other by edges. In each tree collection, we have one root node, which is the very first node in our tree.  
**Graph:** A graph is a non-linear data structure consisting of data elements (finite set) called nodes/vertices and edges that are lines that connect any two nodes in that graph. Each element or node can contain information like roll number, name of student, marks, etc.

**Q.20 What is the difference between tree and graph data structure?**

**Ans.** Both tree and graph data structures are non-linear. Following chart will illustrate the difference in detail:

Tree	Graph
It has only one path between vertices	More than one path is allowed
It has exactly one root node	No root node is available
No loops are permitted	It has loops
It has a hierarchical model	It has a network model
It is less complex	It is more complex
Represents data in tree form	Represents data in network form

**Q.21 How many types are there for Graph data structure? Describe briefly**

**Ans.** A graph data structure is a non-linear type of data structure. It is further divided into two major types:

- Undirected Graph
- Directed Graph

**Undirected Graph:**

It is a type of graph data structure in which nodes are connected by edges that are bidirectional. For example, if an edge connects node 1 and 2, we can traverse from node 1 to node 2, and from node 2 to 1.

**Directed Graph:**

It is a type of graph data structure in which nodes are connected by directed edges that are unidirectional. In this type of data structure, we can only traverse from node 1 to node 2, but not in the opposite direction.

**Q.22 What is a binary tree?**

**Ans.** A binary tree is a special data structure used to store data in which each node can have a maximum of two children. Each node element may or may not have child nodes.



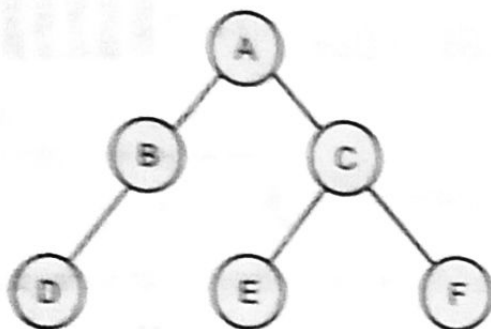
Q.23

Draw the following structure?

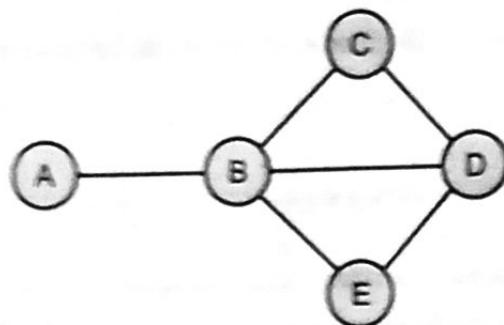
- Tree with six nodes
- Graph with five nodes

Ans.

Tree with six nodes



Graph with five nodes



Q.24 Describe stack and Queue?

**Ans.** **Stack:** Stack is a linear data structure which follows a particular order to perform different operations. Items may be added or removed only at the top of stack. The order may be LIFO (Last In First Out) or FILO (First In Last Out). The term push is used to insert a new element into the stack and pop is used to remove an element from the stack.

**Queue:** A Queue is a linear data structure, which follows a particular order in which operations are performed in FIFO (First In First Out) method, which means that element inserted first will be removed first. The process to add an element into queue is called Enqueue and the process to remove an element from queue is called Dequeue.

Q.25 What is the difference between queue and stack data structure?

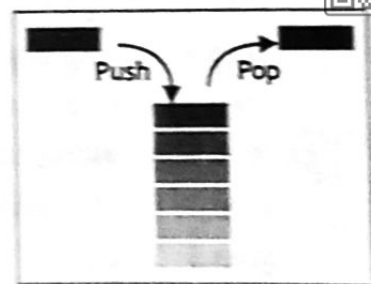
**Ans.** Both queue and stack data structure are type of linear data structure and the difference is explained below:

Stack	Queue
It is based on LIFO principle	It is based on FIFO principle
Alteration of data is done at one end	Alteration of data is done at opposite end
Insert is called as push	Insert is called as enqueue
Deleting data is called as pop	Deleting data is called as dequeue
It uses one pointer	It used two pointers
It has no variants	It has several variants

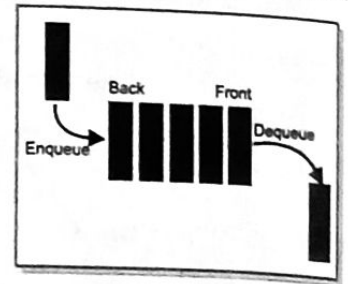
Q.26 Define the terms: Push, Pop, Overflow and Enqueue, Dequeue?

Ans.

- **Push:** It means to enter an item in a stack, so that it becomes the starting point of the data structure.
- **Pop:** It means to remove an item from the stack, moving the rest of the items in a stack one level up.



- **Overflow:** It is an error in a data structure due to shortage of space. A stack overflow is an undesirable condition in which a computer program tries to use more space than available in a stack.
- **Enqueue:** It means to add an element to the queue. It is like push in stack.
- **Dequeue:** It means to remove an element from the front in a queue.



**Q.27 Define array and explain briefly traversal & sorting in array?**

**Ans.** Array is a linear data structure, which holds a list of finite data elements of same data type. Each element of array is referenced by a set of index of consecutive numbers. The elements of array are stored in successive memory locations.

Memory Location									
200	201	202	203	204	205	206	■	■	■
U	B	F	D	A	E	C	■	■	■
0	1	2	3	4	5	6	■	■	■
						Index			

**Traversing Array:** This mode of operation means to access each element of array in order to check data.

**Sorting Array:** An array is said to be sorted when it follows certain order such as numerical, alphabetical etc.

**Q.28 What is the need of index in an array?**

**Ans.** Each location of an element in an array has a numerical value called index. Index are used to identify elements within an array. Indexed means that the elements has been assigned a number in the location which is consecutive.

**Q.29 Draw flowcharts for all the algorithms available in textbook:**

**Algorithm # 1:**

**Step 1:** Start

**Step 2:** Place the fresh water in a pot or a kettle.

**Step 3:** Boil the water.

**Step 4:** Put the black tea leaves in that pot.

**Step 5:** After that add some milk into that pot.

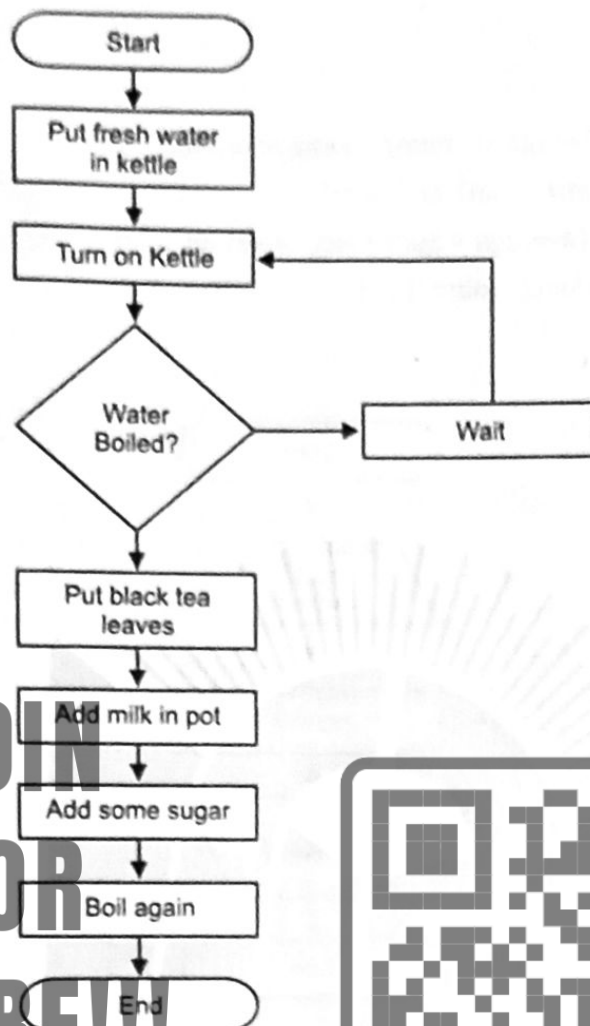
**Step 6:** Add some sugar.

**Step 7:** Boil for some time.

**Step B:** Stop



Ans.



#### Algorithm # 2:

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: Read values num1 and num2.

Step 4: Add num1 and num2 and assign the result to sum.

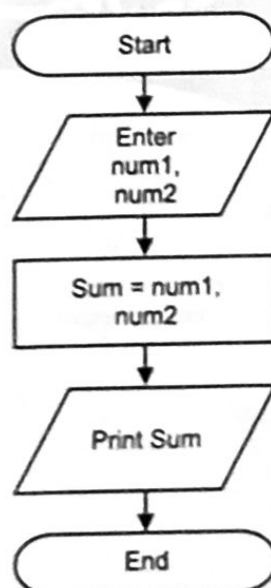
$sum = num1 + num2$

Step 5: Display sum

Step 6: Stop

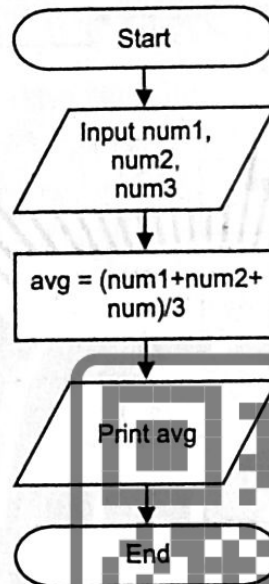


Ans.

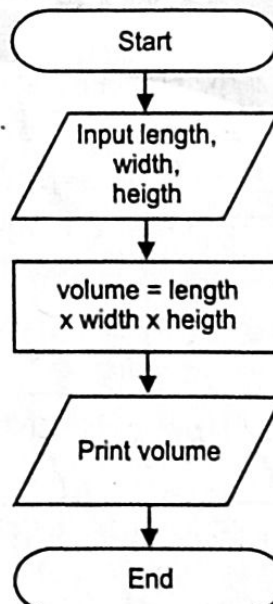


**Algorithm # 3:****Step 1:** Start**Step 2:** Declare variables num1, num2, num3 and avg.**Step 3:** Read values num1, num2 and num3.**Step 4:** Apply formula {Average = Sum / No. of values}

$$\text{avg} = (\text{num1} + \text{num2} + \text{num3}) / 3$$

**Step 5:** Display avg**Step 6:** Stop**Ans.****Algorithm #4:****Step 1:** Start**Step 2:** Declare variables length, width, height and volume**Step 3:** Read values length, width and height.**Step 4:** Apply formula {Volume = length x width x height}

$$\text{volume} = \text{length} \times \text{width} \times \text{height}$$

**Step 5:** Display volume**Step 6:** Stop**Ans.**



### Algorithm # 5:

Step 1: Start

Step 2: Declare variables part, total and percentage.

Step 3: Read values part and total.

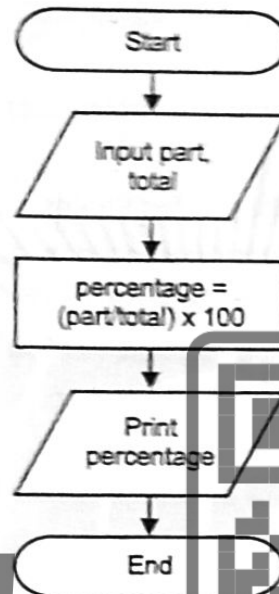
Step 4: Apply formula {Percentage = (part/total) x 100}

percentage = (part/total) x 100

Step 5: Display percentage

Step 6: Stop

Ans.



JOIN  
FOR  
MORE!!!

### Summary

- Problem solving is the process of finding solutions of difficult or complex issues.
- A problem is a situation preventing something from being achieved.
- There are four basic steps involved in finding a solution; define the problem, generate alternative solutions, evaluate and select an alternative and implement and follow up on the solution.
- It is important to understand the problem and set a starting point of solution.
- There are various strategies that can be used to formulate an algorithm for solving the problem.
- Using the strategy, various solutions to a given problem are planned and the most feasible solution is identified.
- Algorithm is a technical term for a set of instructions for solving a problem or sub problem.
- Algorithms enable breaking down of problems and conceptualize solutions step-by step.
- Algorithms are defined as generic steps of instructions so they can be written in any programming language.



- A flowchart writes the sequence of steps and logic of solving a problem, using graphical symbols.
- Flowcharts help in communicating the logic of a program to all others and make it easy for understanding.
- A data structure is a particular way of organizing data in a computer to use it effectively.
- In Linear data structure data elements are arranged in sequential order and each of the elements is connected to its previous and next element.
- Stack is a linear data structure in which items may be added or removed only at one end i.e. at the top of stack.
- Queue is a linear data structure in which that element inserted first will be removed first.
- Array is a linear data structure, which holds a list of finite data elements of same data type. Each element of array is referenced by a set of index of consecutive numbers.
- The elements of a non-linear data structure are not connected in a sequence. Each element can have multiple paths to connect to other elements.
- Tree non-linear data structure is used to represent data containing a hierarchical relationship between elements.
- A graph is a non-linear data structure consisting (finite set) of data elements called nodes/vertices and edges that are lines that connect any two nodes in that graph.

## Solution of Textbook Exercise

### A. Encircle the correct answer:

Ans. See "Multiple Choice Question (MCQ's)" Q.1-10

### B. Respond the following:

1. Describe the steps involved in problem solving.

Ans. See "Short/Detailed Questions and Answers"-Q.2

2. What are the advantages of developing algorithms?

Ans. See "Short/Detailed Questions and Answers"-Q.4

3. List any three advantages of designing flowcharts.

Ans. See "Short/Detailed Questions and Answers"-Q.12

4. What is the difference between tree and graph data structure?

Ans. See "Short/Detailed Questions and Answers"-Q.20

5. What is the difference between queue and stack data structure?

Ans. See "Short/Detailed Questions and Answers"-Q.25

6. What is the need of index in an array?

Ans. See "Short/Detailed Questions and Answers"-Q.28

## Lab Activity

**1.** Design an algorithm to find the greater number by taking two numbers as input.

**Ans.** See "Short/Detailed Questions and Answers" – Q.6

**2.** Design an algorithm to find area of a triangle.

**Ans.** See "Short/Detailed Questions and Answers" – Q.7

**3.** Sort the following steps of the algorithm in correct order for baking a cake:

- Step: Gather the Ingredients
- Step: End
- Step: Grease a pan
- Step: Preheat the oven
- Step: Put the pan in the oven
- Step: Start
- Step: Pour the batter into the pan
- Step: When the timer goes off, take the pan out of the oven
- Step: Set a timer
- Step: Mix together the ingredients to make the batter

**Ans.** See "Short/Detailed Questions and Answers" – Q.8

**4.** Draw a flow chart to calculate gross salary by adding 20% house rent and 30% medical allowances in basic salary.

**Ans.** See "Short/Detailed Questions and Answers" – Q.14

**5.** Draw a flow charts for all the algorithms given in this unit.

**Ans.** See "Short/Detailed Questions and Answers" – Q.29

**6.** Draw the following structure

- a. Tree with six nodes.
- b. Graph with five nodes.

**Ans.** See "Short/Detailed Questions and Answers" – Q.23



UNIT

02

# BASICS OF PROGRAMMING IN C++

JOIN

FOR





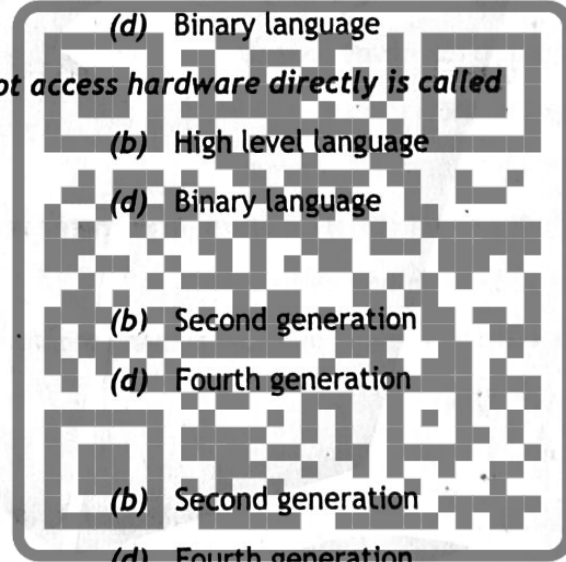
# Multiple Choice Questions (M.C.Qs)

**Choose the right answer:**

- 01.** A computer program is a collection of:  
(a) tasks (b) Instructions (c) Computers (d) Programmers
- 02.** High level languages have syntax that is:  
(a) Easily readable by humans (b) Easily readable by machines  
(c) Easily readable by both (d) None of the above
- 03.** Low level languages have syntax that is:  
(a) Easily readable by humans (b) Easily readable by machines  
(c) Easily readable by both (d) None of the above
- 04.** The primary characteristic of a compiler is to:  
(a) Translate code line by line  
(b) Translate low-level code to machine language  
(c) Detect logical errors  
(d) Translate codes all at once
- 05.** The primary characteristic of an interpreter is to:  
(a) Translate code line by line  
(b) Translate low-level code to machine language  
(c) Detect logical errors  
(d) Translate codes all at once
- 06.** An integrated development environment facilitates a programmer to:  
(a) Edit source code (b) Complete and highlight syntaxes  
(c) Debug and compile codes (d) All of the above
- 07.** All errors detected by users are typically  
(a) Syntax error (b) Semantic error (c) Run-time error (d) Logical error
- 08.** Allowed names for declaring a variable  
(a) Can contain white spaces  
(b) Can be one of the reserved words  
(c) Can contain letters, digits and underscores  
(d) Can be the same as data type
- 09.** A bool data can store following type of values:  
(a) Numbers (b) Strings  
(c) Fractional numbers (d) True or false

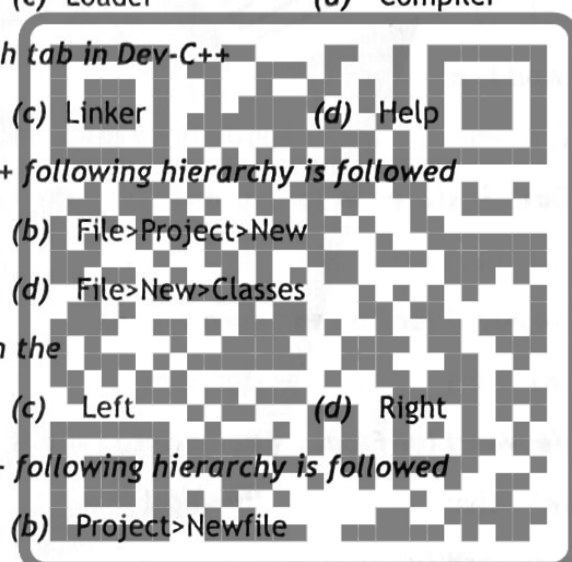


- 10.** Which data type occupies the most space in memory?  
 (a) Character (b) Integer (c) Floating point (d) Double floating point
- 11.** Computer processes instruction in  
 (a) Decimal language (b) Binary language  
 (c) Octal Language (d) Hexa language
- 12.** Programming languages have a specific set of words called  
 (a) data (b) characters (c) syntax (d) pseudocode
- 13.** A programming language that can directly access and communicate with the hardware and is represented by 0 and 1 is called  
 (a) Low-level language (b) High level language  
 (c) Mid-level language (d) Binary language
- 14.** A programming language that can act as a bridge between hardware and user is called  
 (a) Low-level language (b) High level language  
 (c) Mid-level language (d) Binary language
- 15.** A programming language that cannot access hardware directly is called  
 (a) Low-level language (b) High level language  
 (c) Mid-level language (d) Binary language
- 16.** Assemble language belongs to  
 (a) First generation (b) Second generation  
 (c) Third generation (d) Fourth generation
- 17.** Machine language belongs to  
 (a) First generation (b) Second generation  
 (c) Third generation (d) Fourth generation
- 18.** Printing the value of variable without declaring it is known as  
 (a) Syntax error (b) Run-time Error (c) Logical error (d) Spelling error
- 19.** Dividing a number by "0" in a program is considered as  
 (a) Syntax error (b) Run-time Error (c) Logical error (d) Spelling error
- 20.** When desired output is not achieved from a program it is said to possess  
 (a) Syntax error (b) Run-time Error (c) Logical error (d) Spelling error
- 21.** Logical errors always come up with  
 (a) An error message (b) A beep (c) No error message (d) Warning message
- 22.** IDE stands for  
 (a) Implemented design environment (b) Integrated development entry  
 (c) Integrated design environment (d) Integrated development environment

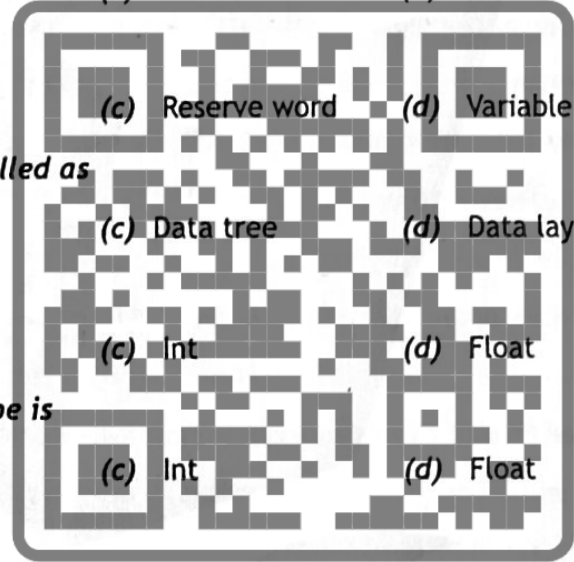


- 23. Eclipse is a**  
 (a) Implemented design environment (b) Integrated development entry  
 (c) Integrated development environment (d) Implemented development environment
- 24. IDE automatically checks for**  
 (a) Output (b) Time (c) Variable (d) Error
- 25. Statements written in a programming language are also called as**  
 (a) Source code (b) Object code (c) Compiler (d) Algorithm
- 26. An IDE can provide**  
 (a) Audio correction (b) Voice recognition (c) Visual clue (d) Virtual assistance
- 27. An operating system program is called as**  
 (a) Compiler (b) Loader (c) Translator (d) Debugger
- 28. Most commonly used IDE for C++ is**  
 (a) Dev-C++ (b) Eclipse (c) Loader (d) Compiler
- 29. Compiler options are available under which tab in Dev-C++**  
 (a) Settings (b) Tools (c) Linker (d) Help
- 30. In order to write a new program in Dev-C++ following hierarchy is followed**  
 (a) File>New>Project (b) File>Project>New  
 (c) File>New>EmptyProject (d) File>New>Classes
- 31. In Dev-C++ project explorer is available on the**  
 (a) Top (b) Bottom (c) Left (d) Right
- 32. In order to add file to a project in Dev-C++ following hierarchy is followed**  
 (a) File>Newproject (b) Project>Newfile  
 (c) Project>addfile (d) File>addproject
- 33. Which shortcut key is used to compile a program in Dev-C++**  
 (a) F10 (b) F9 (c) F11 (d) F5
- 34. Which tab shows the compile status**  
 (a) Execution log (b) Compile log (c) Run log (d) Debug log
- 35. Which shortcut key is used to run a program in Dev-C++**  
 (a) F10 (b) F9 (c) F11 (d) F5
- 36. C++ is a**  
 (a) Special Language (b) General purpose language  
 (c) Dedicated language (d) Software
- 37. C++ was developed by**  
 (a) Elon Musk (b) Bjarne Stroustrup (c) Dennis Ritchie (d) Steve Jobs

**JOIN  
FOR  
MORE!!!**



- 38.** C++ is a  
 (a) Low-level language (b) High-level language  
 (c) Mid level language (d) Dynamic language
- 39.** A reserve word is  
 (a) syntax (b) part of syntax (c) keyword (d) special character
- 40.** Function is an example of  
 (a) Syntax (b) Special character (c) Keyword (d) Program
- 41.** Variable is an example of  
 (a) Reserve word (b) Special character (c) Syntax (d) Program
- 42.** There are total of \_\_\_\_ reserve words in C++  
 (a) 95 (b) 85 (c) 75 (d) 65
- 43.** There are total of \_\_\_\_ reserve words common in C and C++  
 (a) 12 (b) 22 (c) 32 (d) 42
- 44.** Void is a  
 (a) Program (b) Constant (c) Reserve word (d) Variable
- 45.** Data is stored in format and size called as  
 (a) Data Aid (b) Data Type (c) Data tree (d) Data layer
- 46.** Keyword for Boolean data type is  
 (a) Char (b) Bool (c) Int (d) Float
- 47.** Keyword for Floating point data type is  
 (a) Char (b) Bool (c) Int (d) Float
- 48.** Keyword for Integer data type is  
 (a) Char (b) Bool (c) Int (d) Float
- 49.** Keyword for Character data type is  
 (a) Char (b) Bool (c) Int (d) Float
- 50.** Size of Boolean data type is equivalent to  
 (a) 1 bit (b) 1 byte (c) 4 bytes (d) 8 bytes
- 51.** Size of Character data type is equivalent to  
 (a) 1 bit (b) 1 byte (c) 4 bytes (d) 8 bytes
- 52.** Size of Integer data type is equivalent to  
 (a) 1 bit (b) 1 byte (c) 4 bytes (d) 8 bytes
- 53.** Size of Floating point data type is equivalent to  
 (a) 1 bit (b) 1 byte (c) 4 bytes (d) 8 bytes





54. Size of double floating point data type is equivalent to  
 (a) 1 bit (b) 1 byte (c) 4 bytes (d) 8 bytes
55. Range of available character data type in C++ is  
 (a) 0-255 (b) 0-1 (c)  $5.0e^{-345}-1.7e^{308}$  (d)  $1.5e^{-45}-3.4e^{38}$
56. Range of integer data type in C++ is  
 (a) 0-255 (b) -2147483648-2147483647  
 (c)  $5.0e^{-345}-1.7e^{308}$  (d)  $1.5e^{-45}-3.4e^{38}$
57. Range of boolean data type in C++ is  
 (a) 0-255 (b) 0-1 (c)  $5.0e^{-345}-1.7e^{308}$  (d)  $1.5e^{-45}-3.4e^{38}$
58. Range of floating point data type in C++ is  
 (a) 0-255 (b) 0-1 (c)  $5.0e^{-345}-1.7e^{308}$  (d)  $1.5e^{-45}-3.4e^{38}$
59. Range of double floating point data type in C++ is  
 (a) 0-255 (b) 0-1 (c)  $5.0e^{-345}-1.7e^{308}$  (d)  $1.5e^{-45}-3.4e^{38}$
60. A value which cannot be altered during a program is called  
 (a) Integer (b) Constant (c) Variable (d) Boolean
61. A value which can be altered during a program is called  
 (a) Integer (b) Constant (c) Variable (d) Boolean
62. A set of characters which is assigned to a \_\_\_\_\_ variable  
 (a) String (b) Symbolic constant (c) Boolean (d) Source code
63. A name which is given to a constant is called as  
 (a) literal constant (b) Symbolic constant (c) Boolean (d) Source code
64. How many characters can be used in a name using C++  
 (a) 12 (b) 32 (c) 27 (d) 14
65. Which symbol is used to assign value to a variable  
 (a) > (b) + (c) <> (d) =
66. Variable that can store non-numerical values that are longer than one character are called as:  
 (a) Long variable (b) Symbolic Constant (c) Operand (d) Strings

## Answers

- |         |         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|---------|
| 1. (b)  | 2. (a)  | 3. (b)  | 4. (d)  | 5. (a)  | 6. (d)  | 7. (d)  | 8. (c)  |
| 9. (d)  | 10. (d) | 11. (b) | 12. (c) | 13. (a) | 14. (c) | 15. (b) | 16. (b) |
| 17. (a) | 18. (a) | 19. (b) | 20. (c) | 21. (c) | 22. (d) | 23. (c) | 24. (d) |
| 25. (a) | 26. (c) | 27. (b) | 28. (a) | 29. (b) | 30. (a) | 31. (c) | 32. (b) |
| 33. (b) | 34. (b) | 35. (a) | 36. (b) | 37. (b) | 38. (b) | 39. (c) | 40. (a) |
| 41. (a) | 42. (a) | 43. (c) | 44. (c) | 45. (b) | 46. (b) | 47. (d) | 48. (c) |
| 49. (a) | 50. (b) | 51. (b) | 52. (c) | 53. (c) | 54. (d) | 55. (a) | 56. (b) |
| 57. (b) | 58. (d) | 59. (c) | 60. (b) | 61. (c) | 62. (a) | 63. (a) | 64. (b) |
| 65. (d) | 66. (d) |         |         |         |         |         |         |

## Short & Detailed Answer Questions

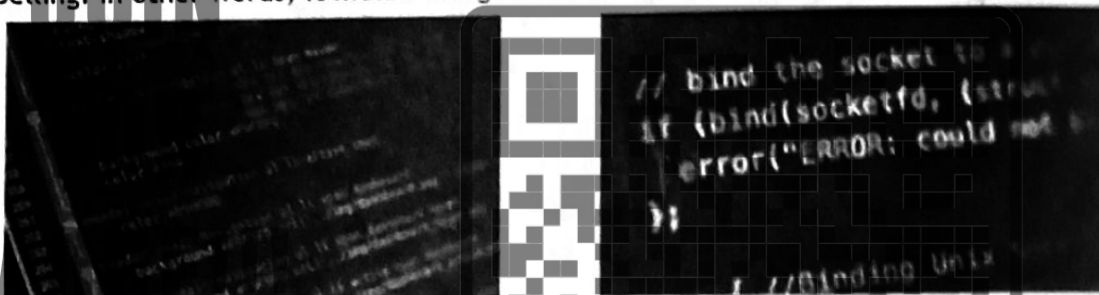
### Q.1 What is a computer program?

**Ans.** A computer program is a set of instructions that is understood by a computer to perform a task. A person who writes a program is known as a programmer.



### Q.2 What is a Syntax in programming language?

**Ans.** When referring to a programming language, the syntax is a set of rules for grammar and spelling. In other words, it means using character structures that a computer can interpret.



### Q.3 Describe the importance of syntax in any computer language?

**Ans.** Syntax tells a computer how to read a set of code. It is essentially a set of keywords and characters that a computer can read, interpret, and convert into a task required.

### Q.4 What are reserved words?

**Ans.** A reserved word in a programming language is a word which has a fixed or predefined meaning which cannot be redefined by a programmer.

### Q.5 Write some reserved words available in C++?

**Ans.** Some reserved words which are available in C++ are `asm`, `bool`, `case`, `char`, `int`, `delete`, `double`.

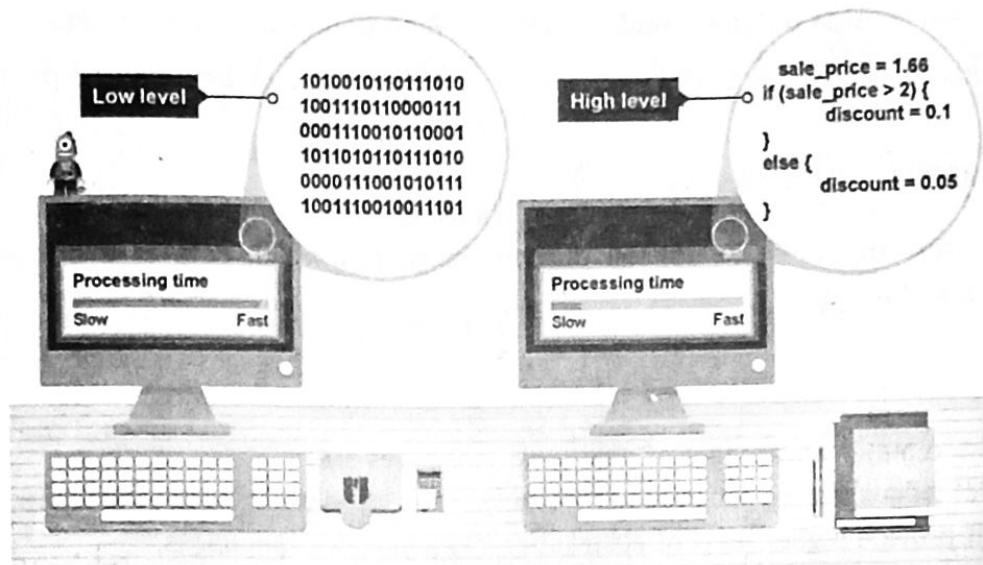
### Q.6 Define a Compiler?

**Ans.** A compiler is a built-in computer program that is used to translate program code written in a high-level computer language (e.g. C++) to a language which a computer can understand.

### Q.7 What are the different types of programming languages?

**Ans.** Each programming language contains a unique set of keywords and syntax, which are used to create a set of instructions. These languages differ in how they communicate with hardware. Following are the three major types of programming languages:

- High Level Language
- Middle Level Language
- Low Level Language

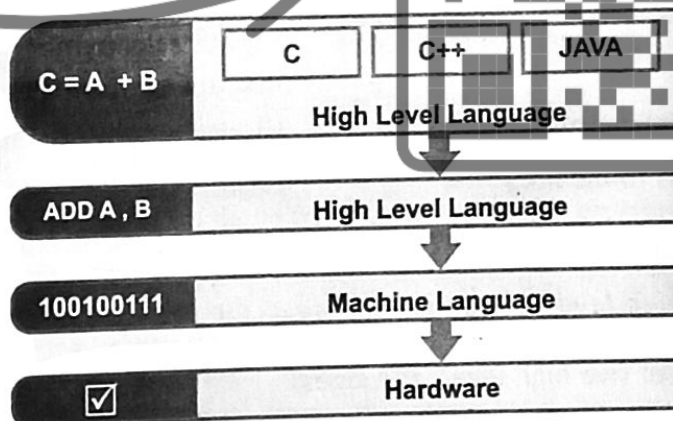


### Q.8 Define High Level Language?

**Ans.** High-level languages allow programmers to write instructions in a language that is easier to understand than low-level languages.

### Q.9 Define Low Level Language?

**Ans.** A low-level language is a type of computer language which is responsible to communicate directly with the hardware. It is represented by 0 and 1. Two common types of low-level languages are machine language and assembly language.



### Q.10 Define Machine Language? OR Define First generation low-level language?

**Ans.** A machine language is a low level language which is written in the form of 0 and 1. Such languages are not easily understood by humans. It is the lowest language in the hierarchy.

### Q.11 Define Assembly Language? OR Define Second generation low-level language?

**Ans.** Assembly language comes above the machine language means it has lesser access to hardware. Assembly language is written in simple English language hence a human can understand it better. Assembler is used to convert source code into machine code.

**Q.12** Write any two differences between machine and assembly language?

Ans.	Low-level language	High-level language
	A program consists of binary i-e 0 and 1	A program consists of syntax
	Only understood by the CPU	Humans can understand the program code

**Q.13** Define Middle level Language?

**Ans.** Middle level languages are special purpose languages which acts as a bridge between hardware and the user.

**Q.14** What are the advantages of high level languages?

**Ans.** Some of the major advantages of high-level languages are:  
High level languages are programmer friendly. They are easy to write, debug and maintain.

- It provide higher level of abstraction from machine languages.
- It is machine independent language.
- Easy to learn.
- Less error prone, easy to find and debug errors.
- High level programming results in better programming productivity.

**Q.15** State and explain difference between low and high level languages?

**Ans.** Some of the major advantages of high-level languages are:

Low-level language	High-level language
It is a machine friendly language i-e operates on 0 and 1	It is a user friendly language and uses English words
It uses assembler to convert assembly code into machine code	It requires compiler or interpreter to convert program into machine code
It is not a portable language	It is not a portable language
It has direct access to memory	It is less memory efficient
Coding and maintenance are complex	Coding and maintenance are easy

**Q.16** List five common high-level languages and describe their purpose?

**Ans.** Following is the list of five high level languages:

**JAVA:**

It is a general-purpose language and possess extreme popularity because of its platform independence. It can run on Windows, Mac, Unix, Linux etc.

**PYTHON:**

It is an interpreted general-purpose high-level language which is famous for its code readability and significant usage of white spaces. Python interpreters are available for multiple platforms and available as open source. It is widely used in information security domain.

**C++:**

It is a general-purpose programming language. It has imperative, object-oriented, and gener



programming features.

#### **SWIFT:**

Swift is a general-purpose, multi-paradigm, compiled programming that was developed by Apple Inc. Swift offers core concepts like dynamic dispatch, late binding, extensible programming, and address errors like null pointer de-referencing, supporting the protocol, extensibility, struts, classes, etc.

#### **RUBY:**

It is an object-oriented high-level language with focus on productivity and simplicity. It is an open source.

### **Q.17 State and explain role of translators in computer language?**

**Ans.** Translator is a computer program that translates a program written in a language into machine code. This is important in order to communicate instructions to the machine.

### **Q.18 Discuss briefly types of translators?**

**Ans.** There are mainly three types of translators which are used to convert computer program into machine equivalent codes:

- Assembler
- Compiler
- Interpreter

#### **Assembler:**

It is a type of translator which is used by assembly language. It has the functionality of compiler and an interpreter. It behaves like a compiler for assembly language but interactive and an interpreter.

#### **Compiler:**

It is a type of translator which is used to convert high-level language to low-level language. Compile generates an executable file by converting the high-level program at once. Because the translation is done in one session an error report is often generated after the complete translation. In order to debug a program the source code requires modification and the executable file will be generated again.

#### **Interpreter:**

It is a type of translator which is used to convert high-level language to low-level language. Compile generates an executable file by converting the high-level program line by line. Interpreter generates error as soon as it is encountered within a source code. This makes it more efficient as compared to a compiler. Interpreters do not create an executable file. Therefore, the interpreter translates the program from the beginning every time it is executed.

### **Q.19 Write down some advantages of Compiler?**

**Ans.** Some advantages of compiler are:

- Source code is not included, therefore compiled code is more secure than interpreted code.
- Tends to produce faster code and interpreting source code.
- Produces an executable file, and therefore the program can be run without need of the source code.



## Q.20 Write down the difference between the compiler and interpreter?

Ans.	Interpreter	Compiler
	Interpreter translates just the statement of the program at a time into machine code.	Compiler translates the entire program and translates the whole of it into machine code at once.
	An interpreter takes very less time to translate the source code.	A compiler takes a lot of time to translate the source code.
	The execution of the program takes place after every line is evaluated and hence the error is raised line by line.	The execution of the program happens only after the entire program is compiled.
	Interpreted program runs slower.	Compiled program runs faster.
	PHP, Ruby are interpreter based languages.	C, C++, Java are compiler based languages.

## Q.21 What is an Error?

Ans. An error is the problem that occurs in a computer program. This causes the program to behave abnormally.

## Q.22 Discuss different types of Error?

Ans. Error are generally classified in three types:

- Syntax Error
- Runtime Error
- Logic Error

### Syntax Error:

Error which occurs in the formation of the computer program are termed as Syntax error. These mistakes vary from misspelled keywords, missing parentheses or misspelled keywords, missing parentheses at incorrect instance. These errors are detected automatically by the computer language and notified with the help of an error message.

### Run-time Error:

Errors which occur during program execution are termed as run-time errors. These errors are encountered after the successful compilation of program. Widely used example is error runtime errors is dividing a number by 0.

### Logic Errors:

Logic errors does not stop a program from getting executed and providing output. However, these errors prevent a program to come up with expected output. Apparently, these program are error free. There are no warnings or error messages attached to logical error.

## Q.23 What is an IDE?

Ans. Integrated development environment (IDE), are applications that facilitates the development of other applications. Developers use numerous tools throughout software code creation building and testing. Development tools often include text editors, code libraries, compiler and test platforms. Without an IDE, a developer must select, deploy, integrate and manage all of these tools separately. An IDE bring many of those development-related tools together in



single framework, application or service. The integrated tool set is designed to simplify software development and can identify and minimize coding mistakes and typos.



**Q.24 Write key benefits of IDE?**

**Ans.** Some key benefits of using IDE are:

- IDE serves as a one window operation for programmers, fulfilling most of the developers need such as compilation, linking, loading and debugging.
- It improves programming workflow.
- Auto-check for errors ensure quality output.
- Refactoring capabilities allow developers to make comprehensive and mistake-free renaming changes.

**Q.25 Discuss components of IDE?**

**Ans.** Components of IDE are discussed below:

**Editing Source Code:**

This is a text editor which is used to write and edit code for a program. Source code editors provide writing and editing facilities to programmers unlike text editors.

**Syntax Highlighting:**

IDE with knowledge of code provides color coding for keywords for a specific language. Highlighting syntax makes code easier to read and visually comfortable.

**Code Completion:**

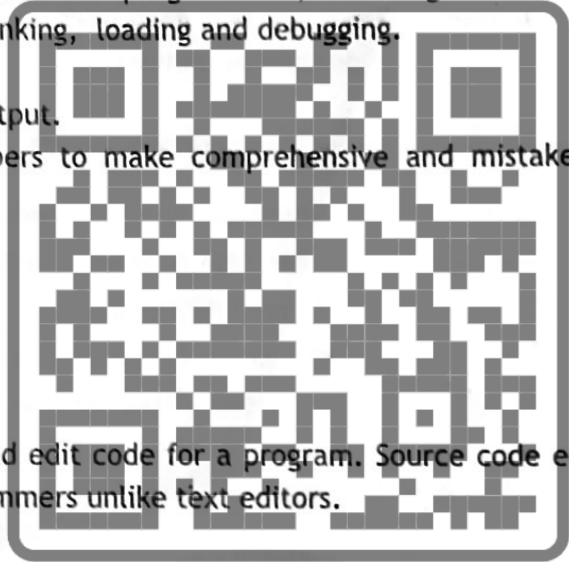
Smart IDEs which are aware of the programming language has the capability of anticipating what a programmer will be writing next. This feature saves time and improves program efficiency.

**Compiler:**

IDEs have built in processes for the translation of programs into machine code. This ensures that program compilation and execution is done automatically.

**Linker:**

Once you have the executable file it is important to identify the source code if you are opening the program again. Linker helps a programmer search code library and links it with the reference code. Unless all linker items are resolved, the process stops and returns the user to the source code file within text editor.



**Loader:**

IDE directs the operating system program called the loader to load the executable file in computers memory and have CPU start processing instructions.

**Debugging:**

A computer program will have errors and it is important to find and debug these errors in order to get required output. IDE provide efficient tools to the programmer to identify and debug errors within a program.

**Q.26 List any four advantages of using an IDE ?**

**Ans.** Following are the advantages of using an IDE:

- Auto-completion of keywords
- Highlighting of syntax
- Auto compilation in
- It reduces human effort and saves time

**Q.27 What are reserved words in C++?**

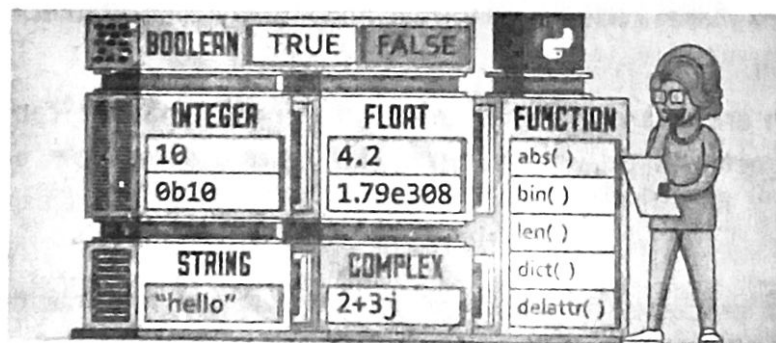
**Ans.** A reserved word in C++ is a word whose meaning is already defined by the compiler. A reserved word cannot be used as an identifier, such as the name of the variable, function or label.

**Q.28 Write down any five reserved words with their functionality?**

- Ans.**
- Bool:** A Boolean data type which can hold true and false.
  - Double:** Basic data type used to define floating number.
  - Delete:** Memory deallocation operator.
  - New:** Memory allocation operator.
  - Void:** Absent of a function parameter list.

**Q.29 What is a data type?**

**Ans.** A data type is a classification that specifies the type of value. A computer program can have data in various types such as Boolean, character, integer etc. Each group is termed as data type.

**Q.30 Explain some common data types used in C++?**

**Ans.** Some common data types in C++ are listed below:

Type	Keyword	Size
Boolean	Bool	1 byte
Character	Char	1 byte



Integer	Int	4 byte
Floating point	Float	4 byte
Double Floating point	Double	8 byte

### Q.31 Define Constant?

**Ans.** A constant is a value that cannot be altered by the program during normal execution, i.e., the value is constant. When associated with an identifier, a constant is said to be "named," although the terms "constant" and "named constant" are often used interchangeably.



### Q.32 Define variable?

**Ans.** A variable is a memory location that can hold a value. Contrary to the constant a value assigned to variable can be changed during execution of the program..

### Q.33 In how many ways a constant is used?

**Ans.** Constants are used in two ways:

- Literal Constant
- Defined Constant

#### Literal Constant:

A literal constant is a value you type into your program wherever it is needed. Examples include the constants used for initializing a variable and constants used in lines of code.

#### Defined Constant:

These are predefined value which can be used within a program. For example, value "Pi" is a defined constant.

### Q.34 Write down difference between variable and constant?

Constant	Variable
A constant does not change its value during program	A variable changes its value depending on the instructions
Constant are usually written as numbers and may be defined as identifiers	Variable are written in letters and symbols
Constants usually represents known values in an equation	Variable represents unknown value in an equation
Constants cannot be redefined	Variables can be redined

### Q.35 Write down rules for naming variables in C++?

**Ans.** General rules for naming variable are:

- Names can contain multiple data types including letters, digits and underscore.
- Name must begin with a letter or underscore.
- Names are case sensitive (MyName and myname are different variables).
- Whitespaces and special characters are not allowed.

- Maximum character length should not exceed 32 characters.

### Q.36 What are strings in C++?

**Ans.** One of the most useful data types supplied in the C++ libraries is the string. A string is variable that stores a sequence of letters or other characters, such as "Hello" or "May 10th is my birthday!".

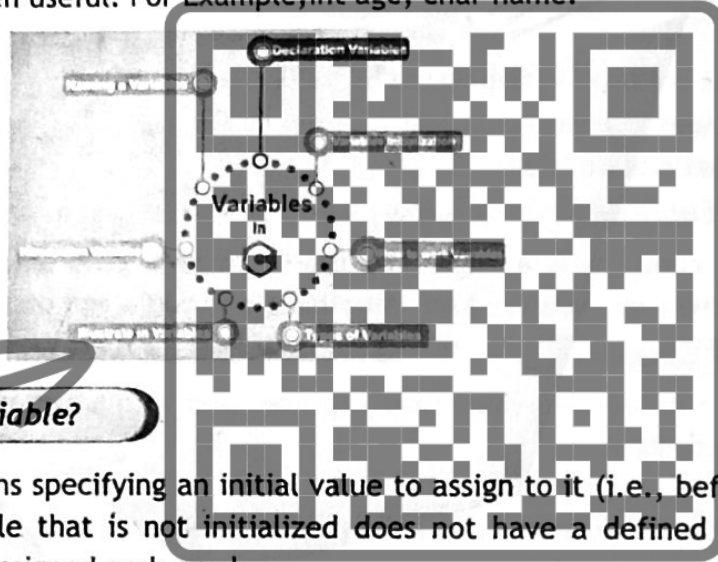
### Q.37 Using the rule for naming variables develop ten meaningful and valid variable names?

**Ans.** Valid variable names are:  
Age, \_age, Name, Employee code, Year\_2, Dateofbirth, Emploment status, surname, \_number, address2.

### Q.38 What is declaring a variable?

**Ans.** Declaring a variable means defining its type, and optionally, setting an initial value (initializing the variable). Variables do not have to be initialized (assigned a value) when they are declared, but it is often useful. For Example, `int age`, `char name`.

JOIN  
FOR  
MORE!!!



### Q.39 What is initializing a variable?

**Ans.** Initializing a variable means specifying an initial value to assign to it (i.e., before it is used at all). Notice that a variable that is not initialized does not have a defined value, hence it cannot be used until it is assigned such a value.

### Q.40 What is the difference between declaring and initializing a variable?

**Ans.** Declaration of a variable is a statement used to identify variable name and its type. It describes the type and location of the value within the program. After declaring it is also important to initialize a variable.

On the other hand, initializing a variable means to assign a value to a variable. The process of initializing a variable is different for different languages.

### Q.41 Write down the difference between source code and object code?

Source Code	Object Code
Created by the programmer	Created by compiler
It is written in human friendly language Human Readable	It is converted in machine friendly language Machine Readable
It is not hardware specific	It is hardware specific

It is the input to the compiler	It is the output from the compiler
It is written in high level language	It is usually written in low level language

## Summary

- A computer program is a list of instruction that tells a computer what to do.
- We refer to syntax in computer programming as the concept of giving specific word sets in specific orders to computers so that they do what we want them to do.
- Different programming languages can be classified as high, middle and low-level languages.
- High-level languages are easy to read for humans and contains English language like words.
- Middle-level languages have a human readable format along with direct control over the machine's resources.
- Low-level languages are easy for machines to read and hard for humans. Low-level programs mostly comprise of binary digits and memory operators.
- There are three types of translators namely compilers, interpreters, and assemblers.
- Compilers convert high-level languages into machine readable format.
- Interpreters also convert high-level languages into machine readable format.
- Unlike compilers, interpreters convert instructions line by line.
- Assemblers convert low-level languages into machine readable format with added benefit of being interactive like an interpreter.
- Programming errors prevents a program from being compiled or executed.
- Syntax errors are words or symbols unrecognized by a particular programming language.
- Run time errors only occur during program execution mostly due to an invalid input.
- Logical errors are considered when incorrect results are obtained based on provided input.
- Logical errors do not interrupt program execution.
- Integrated development environment (IDE) are programs that facilitate writing, compiling and executing codes.
- IDE usually provides a single environment for programmers to write and execute codes efficiently.
- C++ is general-purpose high-level programming language.
- Reserved words are part of programming language syntax and cannot be used a name of variable, function or label.
- A constant is a named identifier having a value that cannot be changed.
- A variable is a named identifier with a value that can be changed during a normal execution of program.
- Different types of values can be stored in variables. These types are called as data types such as int, string, bool, etc.
- A variable can be declared by giving it a name and a type. It can also be initialized during declaration by assigning a value to it.

- In C++ a variable is defined and initialized as :
- "data\_type variable\_name=value;"
- C++ offers various data types for holding values in variables.
- These data types allocate system memory based on its type.

## Solution of Textbook Exercise

### A. Encircle the correct answer:

Ans. See "Multiple Choice Question (MCQ'S)" Q.1-10

### B. Respond the following:

1. What is a computer program?

Ans. See "Short/Detailed Questions and Answers"-Q.1

2. List five common high-level languages and describe their purpose?

Ans. See "Short/Detailed Questions and Answers"-Q.16

3. Using the rule for naming variables develop ten meaningful and valid variable names?

Ans. See "Short/Detailed Questions and Answers"-Q.37

4. Write any two differences between machine and assembly language?

Ans. See "Short/Detailed Questions and Answers"-Q.12

5. What are strings in C++?

Ans. See "Short/Detailed Questions and Answers"-Q.36

6. What is the difference between declaring and initializing a variable?

Ans. See "Short/Detailed Questions and Answers"-Q.40

7. What is the difference between source code and object code?

Ans. See "Short/Detailed Questions and Answers"-Q.41

8. List any four advantages of using an IDE?

Ans. See "Short/Detailed Questions and Answers"-Q.26





**UNIT**  
**03**

**INPUT / OUTPUT  
HANDLING IN C++**



**JOIN  
FOR  
Programming  
MORE..**



# Input & Output in C++

```
#include <iostream>  // header
int main()           // main function
{
    // Standard output object
    // Output operator
    // manipulator
    std::cout << "Enter two numbers:" << std::endl;

    // Standard input object
    // String literal
    // Input operator
    int n1 = 0, n2 = 0;
    std::cin >> n1 >> n2;

    // Variables declarations
    std::cout << "The sum of " << n1 << " and " << n2
    << " is " << n1 + n2 << std::endl;

    // Return statement
    return 0;
}
```

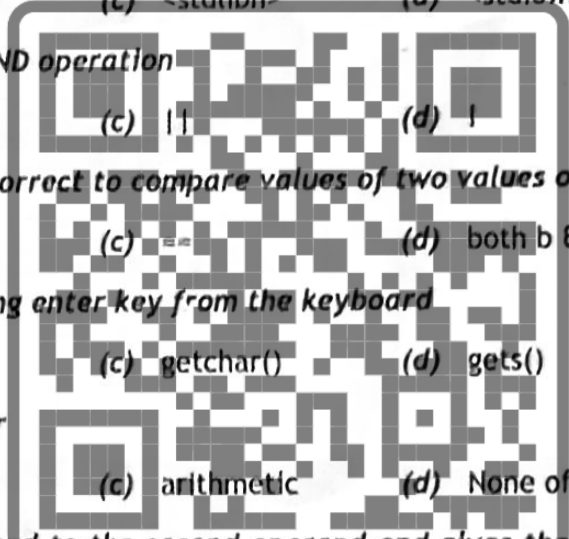
**C++**



## Multiple Choice Questions (M.C.Qs)

Choose the right answer:

1. The C++ header file \_\_\_\_\_ contains function prototype for the standard input and output function.  
(a) <iomain.h> (b) <iostream> (c) <iostream.h> (d) <stdio.h>
2. Which operator is used to input stream  
(a) > (b) << (c) >> (d) <
3. gets stands for  
(a) get stream (b) get string (c) get str (d) get std
4. getch() and getche() are included in \_\_\_\_\_ header file  
(a) <stdio.h> (b) <conio.h> (c) <stdlib.h> (d) <stdio.h>
5. which operator is used for logical AND operation  
(a) & (b) && (c) || (d) |
6. Which of the following operator is correct to compare values of two values of variable  
(a) = (b) <= (c) == (d) both b & c
7. Which of the following needs pressing enter key from the keyboard  
(a) getch() (b) getche() (c) getchar() (d) gets()
8. != belongs to which type of operator  
(a) relational (b) logical (c) arithmetic (d) None of these
9. which operator adds the first operand to the second operand and gives the result to the first operand  
(a) \*= (b) += (c) ++ (d) +
10. cout << 12-6/2; what will be the result of screen?  
(a) 3 (b) 6 (c) 9 (d) 12
11. C++ program is divided into \_\_\_\_\_ parts  
(a) 1 (b) 2 (c) 3 (d) 4
12. The # symbol is called as  
(a) Special character (b) Preprocessor directive  
(c) Program character (d) String
13. #include is used to link  
(a) Internal library (b) external library (c) Standard library (d) prototype library



14. **#define** is used to **DEFINE**

- (a) variable                      (b) constant                      (c) string                      (d) function

15. **Namespace** is a collection of

- (a) words                      (b) constant                      (c) identifiers                      (d) variables

16. **int main(void)** is used for

- (a) compilation                      (b) debugging                      (c) execution                      (d) declaration

17. \_\_\_\_\_ symbol is used at the beginning of the function

- (a) {                      (b) (                      (c) {{                      (d) ((

18. Instructions that performs a particular task is called a

- (a) Function                      (b) Stream                      (c) syntax                      (d) statement

19. \_\_\_\_\_ symbol is used to end the statement

- (a) :                      (b) //                      (c) ;                      (d) —

20. Data type for the main function is

- (a) character                      (b) string                      (c) integer                      (d) float

21. \_\_\_\_\_ symbol is used at the end of the function

- (a) }                      (b) )                      (c) }}                      (d) ))

22. \_\_\_\_\_ statements are ignored by the compiler

- (a) comment                      (b) remarks                      (c) instruction                      (d) errors

23. There are \_\_\_\_\_ types of comment statements

- (a) 1                      (b) 2                      (c) 3                      (d) 4

24. Single line statements are started with

- (a) ??                      (b) //                      (c) ||                      (d) [[

25. double line statements are started with

- (a) \*/                      (b) /\*                      (c) /\*\*                      (d) \*/\*

26. **cout** is used to

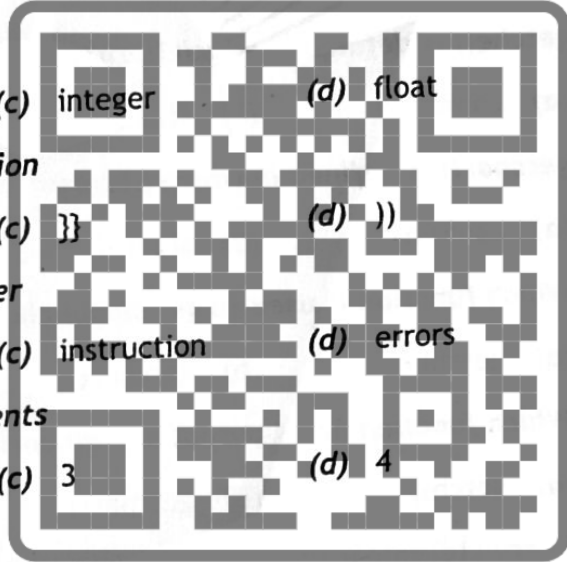
- (a) print output                      (b) manipulate output  
(c) debug output                      (d) compile output

27. **puts()** is used to

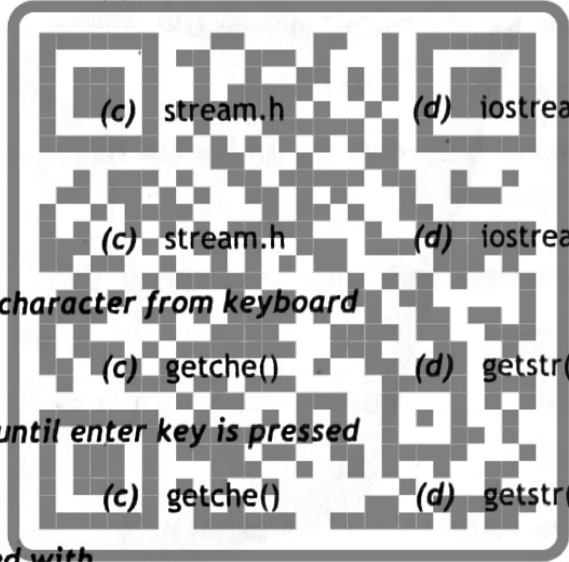
- (a) print output                      (b) manipulate output  
(c) debug output                      (d) compile output

28. **cout** stands for

- (a) character output                      (b) common output  
(c) combine output                      (d) compile output

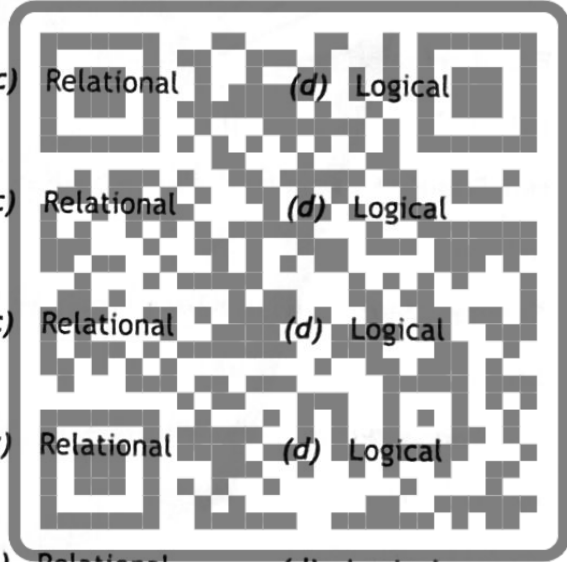


- 29.** *cin* is used to get data from  
 (a) library (b) keyboard (c) voice (d) file
- 30.** *following* operator is used with *cin* to get input  
 (a) == (b) > (c) >> (d) =
- 31.** In a program Console input an output header file is defined as  
 (a) *cstdio.h* (b) *csonio.h* (c) *conio.h* (d) *ioconio.h*
- 32.** *getch()* stands for  
 (a) get compare (b) get character (c) get correct (d) get character echo
- 33.** *getche()* stands for  
 (a) get compare (b) get character (c) get correct (d) get character echo
- 34.** *getch()* is defined in which library  
 (a) *conio.h* (b) *stdio.h* (c) *stream.h* (d) *iostream*
- 35.** *getche()* is defined in which library  
 (a) *conio.h* (b) *stdio.h* (c) *stream.h* (d) *iostream*
- 36.** *getchar()* is defined in which library  
 (a) *conio.h* (b) *stdio.h* (c) *stream.h* (d) *iostream*
- 37.** Which function is used to get single character from keyboard  
 (a) *getch()* (b) *getchar()* (c) *getche()* (d) *getstr()*
- 38.** Which function holds output screen until enter key is pressed  
 (a) *getch()* (b) *getchar()* (c) *getche()* (d) *getstr()*
- 39.** Every statement in C++ is terminated with  
 (a) ; (b) : (c) ) (d) }
- 40.** Escape sequence is a \_\_\_\_\_ character  
 (a) Printable (b) Non-Printing (c) Comparable (d) special
- 41.** Escape sequence starts with  
 (a) / (b) \\ (c) // (d) \
- 42.** *\a* stands for  
 (a) alarm (b) backspace (c) return carriage (d) backslash
- 43.** *\b* stands for  
 (a) alarm (b) backspace (c) return carriage (d) horizontal tab
- 44.** *\t* stands for  
 (a) alarm (b) new line (c) return carriage (d) horizontal tab

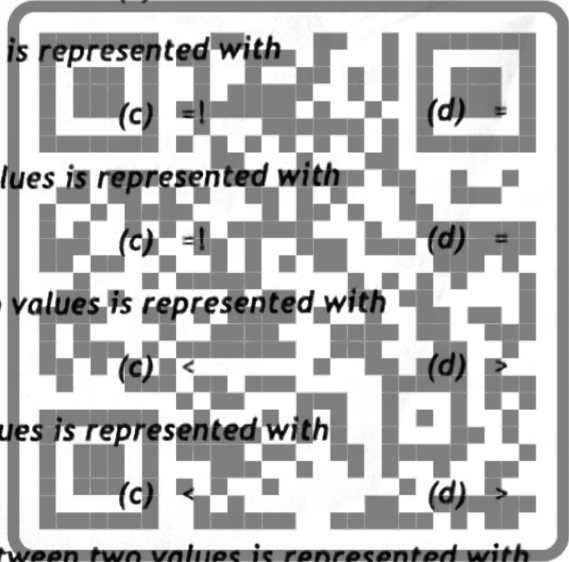




45. \n stands for  
 (a) alarm (b) new line (c) return carriage (d) horizontal tab
46. \r stands for  
 (a) alarm (b) new line (c) return carriage (d) horizontal tab
47. \\ stands for  
 (a) single quotation (b) backspace (c) return carriage (d) backslash
48. \' stands for  
 (a) single quotation (b) backspace (c) double quotation (d) backslash
49. \" stands for  
 (a) single quotation (b) backspace (c) double quotation (d) backslash
50. How many types of operators are used in C++  
 (a) 2 (b) 4 (c) 6 (d) 7
51. + is a \_\_\_\_\_ operator  
 (a) Arithmetic (b) Increment (c) Relational (d) Logical
52. - is a \_\_\_\_\_ operator  
 (a) Arithmetic (b) Increment (c) Relational (d) Logical
53. \* is a \_\_\_\_\_ operator  
 (a) Arithmetic (b) Increment (c) Relational (d) Logical
54. / is a \_\_\_\_\_ operator  
 (a) Arithmetic (b) Increment (c) Relational (d) Logical
55. % is a \_\_\_\_\_ operator  
 (a) Arithmetic (b) Increment (c) Relational (d) Logical
56. Which symbol is used to find remainder of the division  
 (a) / (b) % (c) ^ (d) &
57. When integers are divided “,” output only contains  
 (a) float (b) whole number (c) Decimal number (d) fraction
58. ++ is a \_\_\_\_\_ operator  
 (a) Arithmetic (b) Increment (c) Relational (d) Logical
59. ++ adds \_\_\_\_\_ to the value of variable  
 (a) 1 (b) 2 (c) 11 (d) 0
60. x=++a; is an example of which type of incremental operator  
 (a) prefix (b) postfix (c) single incremental (d) double incremental



61. If  $a=5$  then output for  $x=++a$ ; will be  
 (a) 6 (b) 5 (c) 7 (d) 3
62. If  $a=5$  then output for  $x=a++$ ; will be  
 (a) 6 (b) 5 (c) 7 (d) 3
63.  $--$  is a \_\_\_\_\_ operator  
 (a) decremental (b) Increment (c) Relational (d) Logical
64. If  $a=5$  then output for  $x=--a$ ; will be  
 (a) 6 (b) 5 (c) 4 (d) 3
65. If  $a=5$  then output for  $x=a--$ ; will be  
 (a) 6 (b) 5 (c) 4 (d) 3
66. Result of relational operator is  
 (a) Positive (b) Negative (c) 0 or 1 (d) Whole number
67. Equal relation between two values is represented with  
 (a)  $==$  (b)  $!=$  (c)  $=!$  (d)  $=$
68. Not Equal relation between two values is represented with  
 (a)  $==$  (b)  $!=$  (c)  $=!$  (d)  $=$
69. Greater than relation between two values is represented with  
 (a)  $<<$  (b)  $>>$  (c)  $<$  (d)  $>$
70. less than relation between two values is represented with  
 (a)  $<<$  (b)  $>>$  (c)  $<$  (d)  $>$
71. Greater than equals to relation between two values is represented with  
 (a)  $<=$  (b)  $>=$  (c)  $<$  (d)  $=>$
72. less than equals to relation between two values is represented with  
 (a)  $<=$  (b)  $>=$  (c)  $<$  (d)  $=>$
73.  $\&\&$  logical operator is called as  
 (a) AND (b) OR (c) NOT (d) None of these
74.  $||$  logical operator is called as  
 (a) AND (b) OR (c) NOT (d) None of these
75.  $!$  logical operator is called as  
 (a) AND (b) OR (c) NOT (d) None of these
76. Which operators are used to determine two relational expressions  
 (a) Logical (b) Relational (c) Arithmeti (d) Incrementa



- 77.** Which operators are used to compare two relational statements  
 (a) Logical (b) Relational (c) Arithmetic (d) Incremental
- 78.** Assignment operators is represented by  
 (a) = (b) == (c) - (d) /
- 79.** += operator is called as  
 (a) addition assignment (b) subtraction assignment  
 (c) multiplication assignment (d) division assignment
- 80.** -= operator is called as  
 (a) addition assignment (b) subtraction assignment  
 (c) multiplication assignment (d) division assignment
- 81.** \*= operator is called as  
 (a) addition assignment (b) subtraction assignment  
 (c) multiplication assignment (d) division assignment
- 82.** /= operator is called as  
 (a) addition assignment (b) subtraction assignment  
 (c) multiplication assignment (d) division assignment

## Answers

1. (c)	2. (c)	3. (b)	4. (b)	5. (b)	6. (d)	7. (b)	8. (a)
9. (c)	10. (a)	11. (c)	12. (b)	13. (b)	14. (b)	15. (c)	16. (c)
17. (a)	18. (d)	19. (c)	20. (c)	21. (a)	22. (a)	23. (b)	24. (b)
25. (b)	26. (a)	27. (a)	28. (a)	29. (b)	30. (c)	31. (c)	32. (b)
33. (d)	34. (a)	35. (a)	36. (b)	37. (a)	38. (c)	39. (a)	40. (b)
41. (d)	42. (a)	43. (b)	44. (d)	45. (b)	46. (c)	47. (a)	48. (c)
49. (c)	50. (d)	51. (a)	52. (a)	53. (a)	54. (a)	55. (a)	56. (b)
57. (b)	58. (b)	59. (a)	60. (a)	61. (a)	62. (b)	63. (a)	64. (c)
65. (b)	66. (c)	67. (a)	68. (b)	69. (d)	70. (c)	71. (b)	72. (a)
73. (a)	74. (b)	75. (c)	76. (a)	77. (b)	78. (a)	79. (a)	80. (b)
81. (c)	82. (d)						

## Short & Detailed Answer Questions

**Q.1** Explain basic structure of C++ program?

**Ans.** Basic structure of a C++ program comprises of three components:

- Preprocessor directives
- Main function header
- Body of program

① → # include <iostream>  
② → int main(void)  
③ → {  
④ → Statements;  
⑤ → return 0;  
⑥ → }



**Q.2** What is preprocessor directive?

**Ans.** Preprocessing directives are lines in your program that start with # . The # is followed by a identifier that is the directive name. For example, #define is the directive that defines macro.

**Q.3** What is main function header?

**Ans.** The main function header includes the name of the function and describe what it will expect and return.

**Q.4** Define body of a program in C++?

**Ans.** Body of a program consist of instructions which are enclosed within curly brackets.

**Q.5** What is the use of #include?

**Ans.** #include is used to link the external header libraries which may be required in a program.

**Q.6** What is the use of #define?

**Ans.** It is used to define constants in a program.

**Q.7** What is the use of int main(void) function?

**Ans.** This function is used for execution of C++ program.

**Q.8** Define statement in C++?

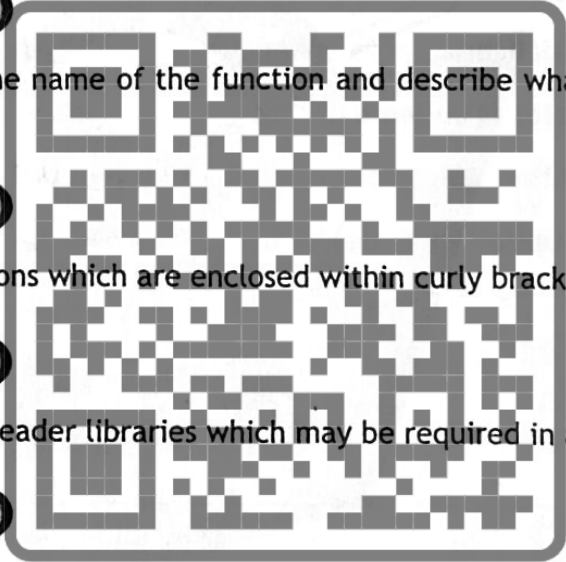
**Ans.** Instructions that perform a particular task is called a statement.

**Q.9** What is the use of terminator (;) in statement?

**Ans.** Terminator is used at the end of a statement in a program.

**Q.10** What is return value?

**Ans.** The return value is the exit code of the program. By default, main() in C++ returns an integer value.





**Q.11** What is a comment statement?

**Ans.** The comment statement are those statements that are ignored by the compiler.

**Q.12** How many types of comment statements do we have?

**Ans.** There are two type of comment statement which are described below:  
Single line Comment: It is used for a single line explanation with the help of double slash (//).  
Double line statement: It is used for the explanation of multiple lines. The explanation should be enclosed with /\* and \*/.

**Q.13** What is an input function? Or Define Input Stream?

**Ans.** In C++ input functions are defined in the form of stream. A function is said to be an input function if the direction of flow of bytes if from the device (Keyboard, mouse, etc) to the main memory.

**Q.14** What is an output function? Or Define output Stream?

**Ans.** In C++ output functions are defined in the form of stream. A function is said to be an output function if the direction of flow of bytes is from the main memory to the device (Monitor, etc).

**Q.15** What is the difference between Input and Output functions?

**Ans.** Following table describe the difference between Input and output functions:

Input Function	Output Function
It is also called as Input stream	It is also called as output stream
Input is taken from devices like keyboard, mouse	Output can be seen on devices like Monitor
Direction of flow of bytes is from device to main memory	Direction of flow of bytes is from main memory to device
Cin is commonly used for taking input	Cout is commonly used for displaying output

**Q.16** Write a C++ to elaborate the use of output function? OR  
Write a C++ which display output "This is my first C++ Program"?

**Ans.**

```
#include <iostream>
using namespace std;
int main() {
    // prints the string enclosed in double quotes
    cout << "This is my first C++ Program";
    return 0;
}
```



**Q.17** Write a C++ to elaborate the use of input function? OR  
Write a C++ program to get integer as input and display on screen?

**Ans.**

```
#include <iostream>
```

```
using namespace std;
int main() {
    int num;
    cout << "Enter an integer: ";
    cin >> num; // Taking input
    cout << "The number is: " << num;
    return 0
}
```

**Q.18** Write the output of the following C++ program? OR  
Write a program to display any three datatypes?

```
#include <iostream>
using namespace std;
int main() {
    int num1 = 70;
    double num2 = 256.783;
    char ch = 'A';
    cout << num1 << endl; // print integer
    cout << num2 << endl; // print double
    cout << "character: " << ch << endl; // print char
    return 0;
}
```

**Ans.** 70  
256.783  
character: A

**Q.19** Write the output of the following C++ program? OR Write a program to get integer and character as input and display both results separately?

```
#include <iostream>
using namespace std;
int main() {
    char a;
    int num;
    cout << "Enter a character and an integer: ";
    cin >> a >> num;
    cout << "Character: " << a << endl;
    cout << "Number: " << num;
    return 0;
}
```

**Ans.** Enter a character and an integer: D 12  
Character: D  
Number: 12

**Q.20** What is the use of PUTS()?

**Ans.** This function is used to print the string to the output stream.

**Q.21 Differentiate between puts() and cout**

Ans.

COUT	PUTS()
Cout is predefined object of OSstream	Puts() is a library function
Cout can print number and string	Puts() can only print string
Header file is iostream.h	Header file is stdio.h
Header file is iostream.h	Puts() does not require insertion operator

**Q.22 Display output "Hello Pakistan" with the help of cout and puts() ?**

Ans.

**COUT:**

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello Pakistan"<<endl;
    return 0;
}
```

**PUTS():**

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main()
{
    puts("Hello Pakistan\n");
    fflush(stdout);
    return 0;
}
```



**Q.23 What is the difference between getch() and getche() ?**

Ans. The difference between getch() and getche() is that, getch() is used to read a single character from the keyboard without buffer and displays value on screen where as getche() is used to read a single character from the keyboard which displays immediately on screen with enter key as it uses buffer.

**Q.24 What is statement terminator?**

Ans. Statement terminator or semi-colon is a symbol which is used to indicate the end of the statement.

**Q.25 Define escape sequence?**

Ans. These are special non-printing characters. An escape sequence starts with a backslash (\) followed by a code character.



**Q.26** Use \a and \r in a single C++ program?

**Ans.**

```
#include<stdio.h>
int main()
{
    printf("Hey this is my first hello world \r\a");
    return 0;
}
```

**Q.27** Write a program that displays the triangle pattern using a single output statement (without using a loop)?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"\n*\n**\n***\n";
}
```

**Q.28** Write a program that displays the arrowhead pattern using a single output statement (without using a loop)?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    cout<<" *\n *\t*\n *\t*\t*\n *\t*\t*\t*";
}
```

**Q.29** Write a program that displays the 5×2 matrix using a single output statement (without using a loop)?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    cout<<" 2\t 4\t 6\t 8\t 10\n 11\t 13\t 15\t 12\t 19 ";
}
```

**Q.30** Write a program that displays square, a cube of a number in table form?

**Ans.**

```
#include <iostream>
using namespace std;
```



```

int main()
{
    cout<<" Number\t Square\t Cube "<<endl;
    cout<<" 1\t"<<1*1<<"\t"<<1*1*1<<endl;
    cout<<" 2\t"<<2*2<<"\t"<<2*2*2<<endl;
    cout<<" 3\t"<<3*3<<"\t"<<3*3*3<<endl;
    cout<<" 4\t"<<4*4<<"\t"<<4*4*4<<endl;
}

```

**Q.31** What is the basic difference between \n and \t?

**Ans.** \t is used to represent tab space and \n is used where a new line is required.

**Q.32** What are operators? Define their types Or Define operators and any specific type?

**Ans.** Operators are special symbols used for specific purpose. There are seven types of operators used in C++:

• **Arithmetic:**

They are used to perform simple arithmetic operations.

• **Incremental:**

++ is an incremental operator which adds 1 to the value.

• **Decremental:**

-- is a decremental operator which deducts 1 from the value.

• **Relational:**

It tests relationship between two values.

• **Logical:**

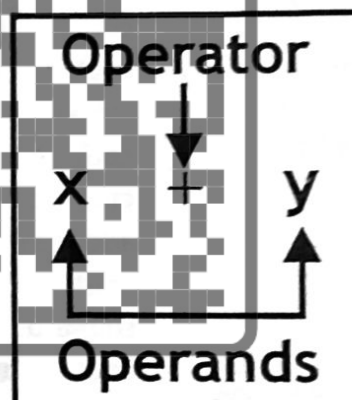
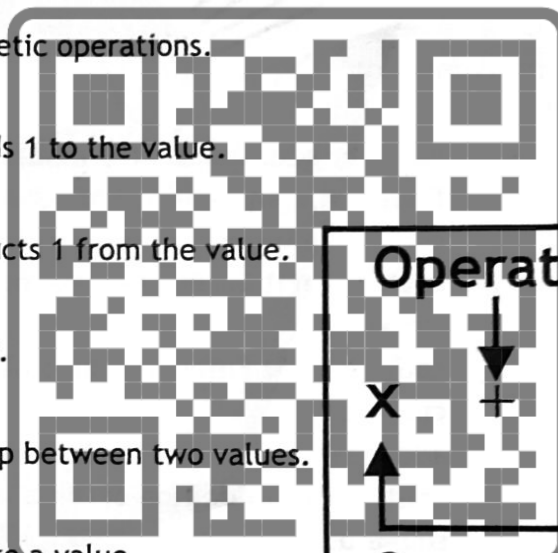
These are used to determine relationship between two values.

• **Assignment:**

This operator is used to assign variable to a value.

• **Arithmetic assignment:**

This is used when equality of values needs to be checked.



**Q.33** Differentiate between arithmetic and relational operators?

Arithmetic Operator	Relational Operator
These operators are used to perform arithmetic operations.	These operators are used to test values.
Arithmetic operations such as addition, subtraction, multiplication etc are performed.	Relations like equal to, not equal to, greater than, less than, etc are performed.
Result of operation is a number.	Result of test is true(1) or false(0).



**Q.34** Differentiate between Logical and relational operators?

Logical Operator	Relational Operator
These operators are used to compare values.	These operators are used to test values.

Logical operators are used to combine or more than one relational expression.	Relations like equal to, not equal to, greater than, less than, etc are performed.
Result of comparison is true(1) or false(0).	Result of comparison is true(1) or false(0).

**Q.35 Differentiate between assignment and equal to operator?**

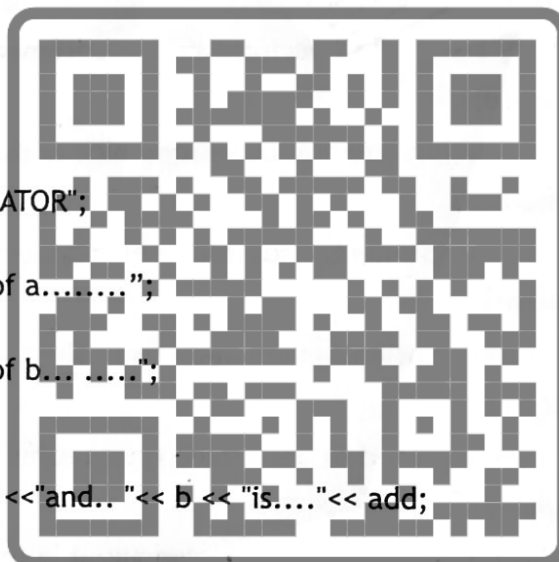
**Ans.**

Assignment Operator	Equal to Operator
It is used to assign variable to a value	It is used to equate values
It is represented by the symbol =	It is represented by the symbol ==
Variable on the left side is assigned value from the right side i-e x=10	It compare value on both sides i-e if x=2 & y=2 then x==y is the equal to relation

**Q.36 Develop a simple calculator in C++ using arithmetic values?**

**Ans.**

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main(void)
{
    int a, b, add, sub, multi, remd;
    float div; cout << "\n \t CALCULATOR";
    cout << "\n \t =====";
    cout << "\n \t Enter the value of a.....";
    cin >> a;
    cout << "\n \t Enter the value of b... ..";
    cin >> b;
    add = a+b;
    cout << "\n \t Addition of "<< a << "and.. " << b << "is...." << add;
    sub=a-b;
    cout << "\n \t Subtraction of "<< a << "and..." << b << "is...." << sub;
    multi=a*b;
    cout << "\n \t Multiplication of "<< a << "and..." << b << "is...." << multi;
    div=a/b;
    cout << "\n \t Division of "<< a << "and..." << b << "is...." << div;
    remd=a%b;
    cout << "\n \t Remainder of Modulus division of "< << "and..." << b << "is...." << remd;
    return 0;
}
```



**Q.37 Write a C++ program and use relational operators? OR Show how relational operators can be used in a C++ program?**

**Ans.**

```
#include<iostream>
using namespace std;
int main(void)
```

```

{
    int a = 10;
    int b = 20;
    cout << "\n \t Relational Operator";
    cout << "\n \t =====";
    cout << "\n \t" << "False \t" << (a == b) << "\t false because 10 is not equal to 20";
    cout << "\n \t" << "True \t" << (a < b) << "\t true because 10 is less than 20";
    cout << "\n \t" << "False \t" << (a > b) << "\t false because 10 is not greater than 20";
    cout << "\n \t" << "False \t" << (b <= a) << "\t false because 20 is not less than or equal to 10";
    cout << "\n \t" << "True \t" << (a != b) << "\t true because 10 is not greater than 10 but equal to 10";
    cout << "\n \t" << "True \t" << (b != a) << "\t true because 20 is not equal to 10";
    return 0;
}

```

**Q.38**

**Write a C++ program and use relational and logical operators together? OR Show how relational operators and logical operators can be used in a C++ program?**

**Ans.**

```

#include <iostream>
using namespace std;
int main(void)
{
    int x = 10;
    int y = 5;
    int z = 12;
    cout << "\n \t LOGICAL OPERATOR";
    cout << "\n \t =====";
    cout << "\n \t" << ((x > y) && (x < z)) << "\n \t ADD OPERATOR" << endl;
    cout << "\n \t" << ((x > y) || (x > z)) << "\n \t OR OPERATOR" << endl;
    cout << "\n \t" << !(x < y) << "\n \t NOT OPERATOR" << endl;
    return 0;
}

```



**Q.39**

**Write a C++ program and use assignment and equal to operators together? OR Show how assignment operators and equal to operators can be used in a C++ program?**

**Ans.**

```

#include <iostream>
using namespace std;
int main(void)
{
    int x=20,
    y=10;
    cout << "\n \t Assignment vs Equal to Operator";
    cout << "\n \t =====";
}

```



```
cout << "\n \t x = 20 assignment opt....." << x;
cout << "\n \t y = 10 assignment opt....." << y;
cout << "\n \t Equal to opt. result is....." << (x==y);
return 0;
```

```
}
```

**Q.40** Write Output of the following C++ programs:

**A.** `#include <iostream>`  
`using namespace std;`  
`int main(void)`  
`{`

```
    int a = 27;
    cout << "a is " a < endl;
    cout << "a is now" < a++ < endl;
    cout << "a is now " < a < endl;
    cout << "a is now " << --a << endl;
    cout << "a is now " < a << endl;
    return 0;
```

```
}
```

**Ans.** a is 27  
a is now 28  
a is now 28  
a is now 27  
a is now 27

**B.** `#include<iostream>`  
`using namespace std;`  
`{`

```
    float radius, area;
    cout << "Enter radius of circle: ";
    cin >> radius;
    area = 3.14*radius*radius;
    cout << "Area = " << area << endl;
    return 0;
}
```

**Ans.** Enter radius of circle: 10  
Area = 314

**C.** `#include<iostream>`  
`#include<math.h>`  
`using namespace std;`  
`int main()`  
`{`

```
    float a, b, c, s, area;
    cout << "Enter length of three sides of triangle: ";
    cin >> a >> b >> c;
    s = (a + b + c) / 2;
```





```

area= sqrt( s * (s-a) * (s-b) * (s-c) );
count << "Area = " << area << endl;
return 0;
}

```

**Ans.** Enter length of three sides of triangle: 4.5 8.9 12  
Area = 16.644

#### Q.41 Develop following programs in C++?

**A.** Calculate speed of an object by using distance formula i-e  $s=d/t$

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
int dis,sp;
float t;
cout<<"Enter the distance: ";
cin>>dis;
cout<<"Enter the time: ";
cin>>t;
sp=dis/t;
cout<<"The speed required is: "<<sp<<endl;
}

```

**B.** Calculate force by using  $F=ma$

**Ans.**

```

#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
float f,m,a;;
cout<<"Enter the value of mass in Kg : ";
cin>>m;
cout<<"Enter the acceleration of the body : ";
cin>>a;
f=(m*a);
cout<<"The force according to Newton's second law of motion = "<<f<<" N ";
getch();
}

```

**C.** Calculate acceleration of a body by using formula  $a=(V_f-V_i)/t$

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
float Vf,Vi,a,t ;
cout<<"Enter initial velocity \n" ;
cin>> Vi;

```



```

cout<<" Enter final velocity \n" ;
cin>> Vf ;
cout<<"Enter time \n";
cin>> t ;
a=(Vf-Vi)/t;
cout<<a ; ;
return 0;
}

```

**D.** Find area of triangle by  $a = \frac{1}{2} * l * b$

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
    int height, base;
    float area; //ans may come in fractions
    cout<<"Enter height and base : ";
    cin>>height>>base;
    area= (0.5)*height*base; //area of triangle formula
    cout<<"Area of triangle is : "<<area;
    return 0;
}

```

**E.** Convert temperature from Celsius to Fahrenheit by using  $F = (C * 1.8) + 32$

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
    float fahrenheit, celsius;
    cout << "Enter the temperature in celsius\n";
    cin >> celsius;
    fahrenheit =(1.8 * celsius) + 32;
    cout << celsius <<"Centigrade is equal to " << fahrenheit <<"Fahrenheit";
    return 0;
}

```

**Q.42** Write a program to calculate volume of a box?

**Ans.**

```

#include <iostream.h>
#include <conio.h>
int main()
{
    cout << "Volume of a box Formula: V = l * w * h \n";
    float w, l, h, V;
    cout << "Enter the value of l-length?\n";
    cin >> l;
    cout << "Enter the value of w-Width?\n";
    cin >> w;
    cout << "Enter the value of h-height?\n";
}

```



```

cin >> h;
V = l * w * h;
cout << "\t\tTHE VOLUME OF box\t" << V << " m^3 " << endl;
getch();
return 0;
}

```

**Q.43** Write a program of Marksheets to take input of five subjects, print its total and percentage also?

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
    float p, c, m, e, u, total, percentage;
    // p, c, m, e, and h are the five subjects
    // p = physics
    // c = computer
    // m = math
    // e = english
    // u = urdu
    cout << "Enter the marks of five subjects:\n";
    cin >> p >> c >> m >> e >> h;
    total = p + c + m + e + h;
    percentage = (total / 500.0) * 100;
    cout << "The Total marks = " << total << "/500\n";
    cout << "The Average marks = " << average << "\n";
    cout << "The Percentage = " << percentage << "%";
    return 0;
}

```



**Q.44** Write a code to calculate mathematical expression of  $a^2 + 2ab + b^2$ ?

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
    int a, b, output;
    cout << "Enter a value :";
    cin >> a;
    cout << "Enter b value :";
    cin >> b;
    output = a^2 + 2*a*b + b^2;
    cout << "Output is : " << output << endl;
    return 0;
}

```



**Q.45** Find out errors in the following code?

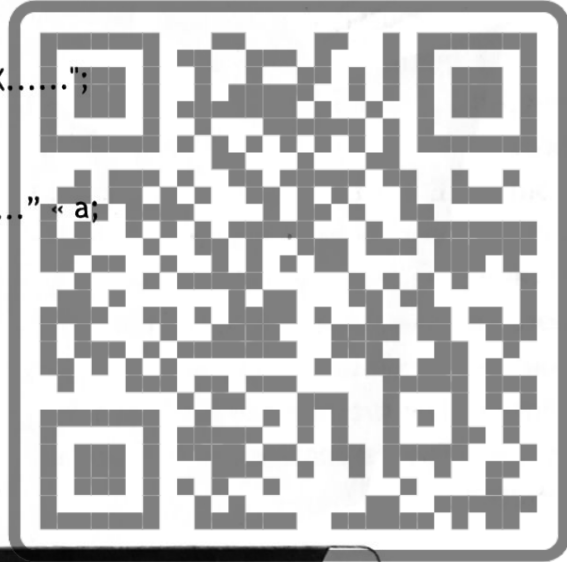
```
#include<iostream.h>
using namespace std
int main(void);
{
    int x;
    cout << "\n Enter the value of X.....";
    cin >> x
    cout << "\n The square of X....." << a * a;
    return 0;
}
```

**Ans:**

```
#include<iostream.h>
using namespace std ;
int main(void);
{
    int x,a;
    cout << "\n Enter the value of X.....";
    cin >> x;
    a=x*x;
    cout << "\n The square of X....." << a;
    return 0;
}
```

**Output:**

Enter the value of X.....2  
The square of X.....4



## Summary

- The C++ program consists of three parts.
- Preprocessor Directives
- main Function header
- Body of program
- `#include<iostream>` is used to include header files like `iostream.h`, `conio.h`, etc.
- namespace is the collection of identifiers.
- The `main()` function is compulsory element of the C++ program
- The comment statements are those statements that are ignored by the compiler.
- These statements are not executable.
- I/O Stream is a standard library file that contains definitions of Standard Input Output functions. `cout` is an output object.





- It is used to display the output through output device like monitor.
- puts() is a string function and it is included in <stdio.h> header file.
- cin works as an input object in C++.
- Statement terminator (;) is used for statement ending in C++ programming
- Escape Sequences are non-printable characters. It is used only with cout statement.
- Operators are special symbols used for specific purposes.
- Arithmetic Operators are used for arithmetic operations or calculation
- Increment and Decrement Operators are used in two different ways in programming.
- Prefix
- Postfix
- Relational Operators are used to test the relation between two values.
- Logical Operators are used to determine two relational expressions. Relational and Logical Operators work on a decision making and loops.

## Solution of Textbook Exercise

### A. Encircle the correct answer:

Ans. See "Multiple Choice Questions (MCQs)" Q.1-10

### B. Respond the following:

1. Use \a and \r both escape sequences in a program?

Ans. See "Short/Detailed Questions and Answers" - Q.26

2. How many types of comment statements are used in C++?

Ans. See "Short/Detailed Questions and Answers" - Q.12

3. Differentiate between arithmetic and logical operators?

Ans. See "Short/Detailed Questions and Answers" - Q.33

4. Write a program in C++ and use all arithmetic operators?

Ans. See "Short/Detailed Questions and Answers" - Q.36

5. What is the basic difference between equal to and assignment operator?

Ans. See "Short/Detailed Questions and Answers" - Q.35

6. What is the basic difference between \n and \t?

Ans. See "Short/Detailed Questions and Answers" - Q.31

7. Get the output of the program?

Ans. See "Short/Detailed Questions and Answers" - Q.40



## Lab Activity

1. Develop programs for manipulating the following formulas:  
Ans. See "Short/Detailed Questions and Answers"– Q41
2. Write a program to calculate volume of a box?  
Ans. See "Short/Detailed Questions and Answers"– Q42
3. Write a program of marksheet takes input of five subjects, print its total percentage also?  
Ans. See "Short/Detailed Questions and Answers"– Q43
4. Write a code to calculate mathematical expression- $a^2+2ab+b^2$   
Ans. See "Short/Detailed Questions and Answers"– Q44
5. List out the errors from the following program, eliminate these errors and write the output.

```
#include<iostream.h>
using namespace std
int x;
```

```
cout << "\n Enter the value of X.....";
```

```
int main(void);
```

```
{
```

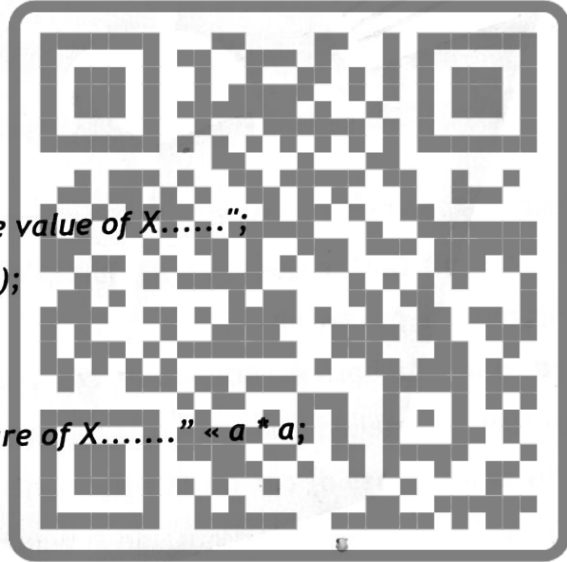
```
cin >> x
```

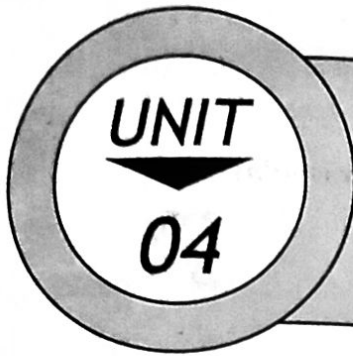
```
cout << "\n The square of X....." << a * a;
```

```
return 0;
```

```
}
```

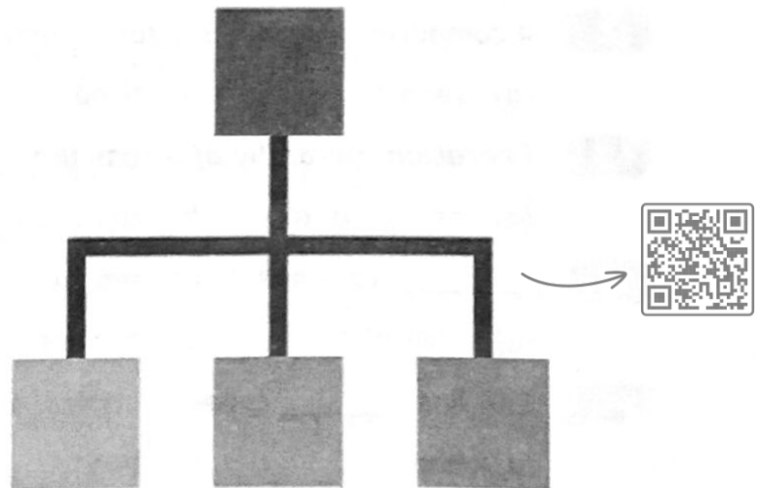
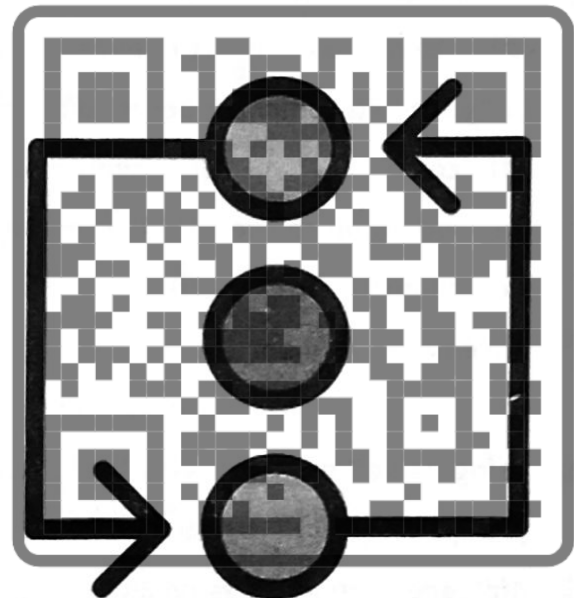
- Ans. See "Short/Detailed Questions and Answers"–Q45





# CONTROL STRUCTURE

**JOIN  
FOR  
MORE!!!**



## Multiple Choice Questions (M.C.Qs)

Choose the right answer:

1. Loop within a loop is called as  
(a) inner (b) outer (c) enclosed (d) nested
2. "Case" and \_\_\_\_\_ are also part of switch statement  
(a) have (b) default (c) for (d) if
3. "for" loop expression has \_\_\_\_\_ parts  
(a) one (b) two (c) three (d) four
4. exit() function is used to  
(a) close function (b) close loop (c) close program (d) close switch
5. "continue" statement takes control to  
(a) top of loop (b) end of loop (c) top of function (d) end of function
6. In "goto" statement label is followed by \_\_\_\_\_ character  
(a) colon : (b) semi colon ; (c) single quotation " (d) double quotes "
7. To send value to the calling function we use \_\_\_\_\_ statement  
(a) through (b) return (c) send (d) back
8. "break" statement is used with \_\_\_\_\_  
(a) if (b) switch (c) for (d) while
9. using "else" is \_\_\_\_\_ with "if" statement  
(a) prohibited (b) advised (c) compulsory (d) optional
10. "if" and loop expression use \_\_\_\_\_ operators to test condition  
(a) arithmetic (b) relational (c) insertion (d) bitwise
11. A computer program is a set of instruction in \_\_\_\_\_ form  
(a) random (b) fixed (c) unstructured (d) sequential
12. Execution Hierarchy of a computer program is from  
(a) bottom to top (b) top to bottom (c) left to right (d) right to left
13. \_\_\_\_\_ statements are used to control direction of the program  
(a) control (b) arithmetic (c) logical (d) relational
14. C++ has \_\_\_\_\_ types of control statements  
(a) one (b) two (c) three (d) four
15. Decision making control structure is divided into \_\_\_\_\_ groups  
(a) one (b) two (c) three (d) four



16. Body of statement is also called as  
 (a) corner (b) safe (c) wall (d) block
17. A loop body is enclosed within  
 (a) () (b) " " (c) [] (d) {}
18. In "if" loop if the statement is true the flow is moved to \_\_\_\_ of the loop  
 (a) top (b) bottom (c) block (d) end
19. In nested "if" inner "if" is tested only when \_\_\_\_ is true  
 (a) outer "if" (b) condition (c) start (d) inner "if"
20. In which loop if the condition is true the execution moves if block and if it is false the program execution moves to "else" block  
 (a) else-if (b) if-else (c) for while (d) if then
21. If-else nested deeply is called as  
 (a) nested if (b) else-if (c) if-if else (d) else-if-else
22. A switch statement can have more than one \_\_\_\_ statement  
 (a) if (b) for (c) case (d) else
23. A case statement is followed by \_\_\_\_ or \_\_\_\_  
 (a) float, character (b) integer, string  
 (c) decimal, character (d) integer, character
24. if switch variable does not match with any of the case constant, control goes to  
 (a) end (b) top (c) default (d) block
25. "Default" in switch statement is  
 (a) prohibited (b) advised (c) compulsory (d) optional
26. A case in switch statement can have \_\_\_\_ operator only  
 (a) < > (b) + (c) = (d) /
27. iteration is also called as  
 (a) loop (b) statement (c) command (d) structure
28. There are \_\_\_\_ types of loop in C++  
 (a) one (b) two (c) three (d) four
29. In order to execute sequence of statements multiple times \_\_\_\_ is used  
 (a) operator (b) statement (c) library (d) loop
30. exit function uses \_\_\_\_ library file  
 (a) stdio.h (b) conio.h (c) stdlib (d) stdx.h
31. first expression in "for" statement is called as  
 (a) initialization (b) test  
 (c) increment/decrement (d) None of the above



32. second expression in "for" statement is called as  
 (a) initialization (b) test  
 (c) increment/decrement (d) None of the above
33. third expression in "for" statement is called as  
 (a) initialization (b) test  
 (c) increment/decrement (d) None of the above
34. "while" loop is also called as  
 (a) preset (b) pretest (c) posttest (d) counter control
35. "do-while" loop is also called as  
 (a) preset (b) pretest (c) posttest (d) counter control
36. which is called as indefinite repetition loop  
 (a) while (b) else (c) for (d) if
37. "for" loop is also called as  
 (a) preset (b) pretest (c) posttest (d) counter control
38. which is called as definite repetition loop  
 (a) while (b) do-while (c) for (d) if
39. which statement is used to change execution sequence of programs  
 (a) logical (b) jump (c) relational (d) arithmetic
40. which jump statement transfer the control immediately from loop or switch statement.  
 (a) break (b) goto (c) continue (d) return
41. which jump statement transfer the control to top by skipping remaining statement body  
 (a) break (b) goto (c) continue (d) return
42. A \_\_\_\_\_ statement transfer control unconditionally to a labelled statement in same function:  
 (a) break (b) goto (c) continue (d) return
43. A \_\_\_\_\_ statement transfer control to statement just after function call  
 (a) break (b) goto (c) continue (d) return
44. \_\_\_\_\_ function is used to terminate C++ program  
 (a) goto (b) exit (c) return (d) break

## Answers

1. (d)	2. (b)	3. (c)	4. (c)	5. (a)	6. (a)	7. (b)	8. (b)
9. (d)	10. (b)	11. (d)	12. (b)	13. (a)	14. (c)	15. (c)	16. (d)
17. (d)	18. (c)	19. (a)	20. (b)	21. (b)	22. (c)	23. (d)	24. (c)
25. (d)	26. (c)	27. (a)	28. (c)	29. (d)	30. (c)	31. (a)	32. (b)
33. (c)	34. (b)	35. (c)	36. (a)	37. (b)	38. (c)	39. (b)	40. (a)
41. (c)	42. (b)	43. (d)	44. (b)				

# Short & Detailed Answer Questions

**Q.1** Define the term control statement?

**Ans.** Statements which are used to control the sequence of a program are termed as control statements.

**Q.2** Discuss all types of control statements? OR  
How many types of control statements are there? Discuss briefly

**Ans.** There are three types of control statements:

- Decision making statement
- Loop
- Jump

**Decision making statement:**

As the names suggest these kinds of statements are used to decide whether certain part of code should be executed or not.

**Loop:**

Loop is also called as iterations. It is used when a certain part of the programs needs to be executed more than once.

**Jump:**

Jump statements are used to change the flow of program. There are several ways to break the execution and change flow of a program depending on the requirement.

**Q.3** How many types of decision-making structures are used in C++?

**Ans.** There are three types of decision-making structures in C++:

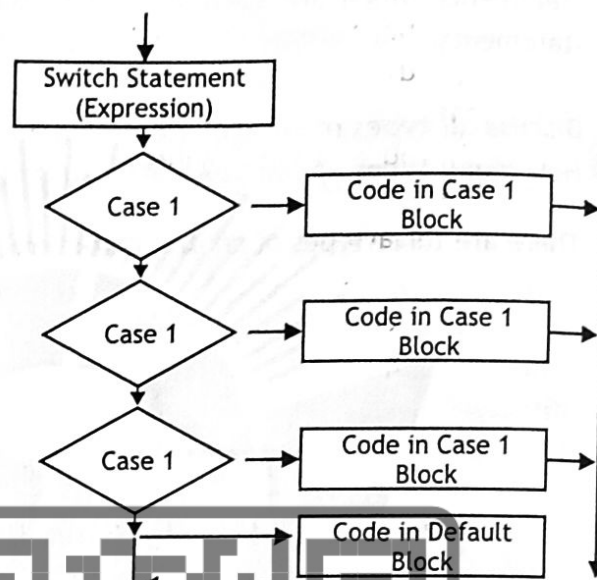
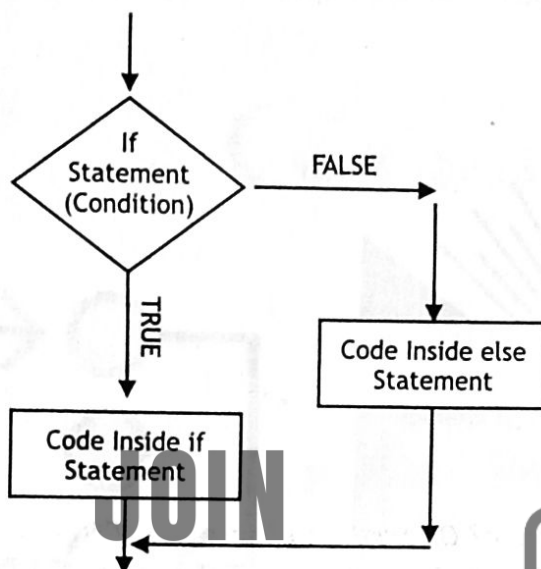
- “If” statement
- “if-else” statement
- “switch” statement

**Q.4** What is the syntax of “if” statement?

**Ans.** if (test condition)- is the syntax of the if statement it checks a condition and decides the execution sequence.

**Q.5 Explain the syntax of "if-else" statement?**

**Ans.** If condition returns true then the statements inside the body of "if" are executed and the statements inside body of "else" are skipped.  
If condition returns false then the statements inside the body of "if" are skipped and the statements inside the body of "else" are executed.



**Q.6 Define syntax of switch statement?**

**Ans.** A typical syntax involves: the first select, followed by an expression which is often referred to as the control expression or control variable of the switch statement. Subsequent lines defining the actual cases (the values), with corresponding sequences of statements for execution when a match occurs.

**Q.7 Differentiate between if and if-else?**

**Ans.**

If statement	If-else statement
It consists of an expression followed by one or more statement	It has a if statement followed by an else statement which is executed when at false
In case the if test is false the sequence jumps outside the block.	In case the if statement is false the sequence jumps to the else block

**Q.8 Differentiate between if-else and switch?**

**Ans.**

Switch statement	If-else statement
switch statement uses single expression for multiple choices.	if-else statement uses multiple statement for multiple choices.
switch statement test only for equality.	if-else statement test for equality as well as for logical expression.
switch statement evaluates only character or integer value.	evaluates integer, character, pointer or floating-point type or boolean type.
If the condition inside switch statement does not match with any of cases, for that instance the default statement is executed if created.	If the condition inside if statement is false, then by default the else statement is executed if created.



**Q.9**

Write a C++ program by using if statement? OR Write a program which takes marks as input. If marks are greater than 60 then it adds word "good" between "you are a" and "student of this class."

**Ans.**

```
#include<stdio.h>
#include <iostream.h>
using namespace std;
int main(void)
{
    int marks; cout<<"\nEnter Marks";
    cin>>marks;
    cout<<" You have secured "<<marks<<" marks ";
    if(marks>=70)
        cout<<" and A Grade ";
    cout<<" in 5th class";
    return 0;
}
```

**Q.10**

Write a C++ program by using Nested-if statement? OR Write a program, which takes marks and age of a person as input. First condition checks marks, if they are greater than or equal to 60 then next condition checks age. If age is also greater than 18 then it prints message "you got the job". "good luck" is always printed.

**Ans.**

```
#include<stdio.h>
#include <iostream.h>
using namespace std;
int main(void)
{
    int marks, age;
    cout<<"\nEnter your Marks ";
    cin>>marks; cout<<"\nEnter your age ";
    cin>>age;
    if(marks>=60)
        if(age>=18)
            cout<<"you got the job";
    cout<<" Good Luck";
    return 0;
}
```

**Q.11**

Write a C++ program by using if-else statement? OR Write a program which takes marks as input and decides pass or fail on the basis of marks. If greater than or equal to 40 then pass otherwise fail.

**Ans.**

```
#include<stdio.h>
#include <iostream.h>
using namespace std;
int main(void)
{
```

```

int marks;
cout<<"\nEnter your Marks";
cin>>marks;
if(marks>=40)
cout<<"You are pass";
else
cout<<" You are fail";
return 0;
}

```

**Q.12** Write a C++ program by using nested if-else or else-if statement? OR Write a program which takes marks as input and then determines the grade by applying if condition multiple times?

**Ans.**

```

#include<stdio.h>
#include <iostream.h>
using namespace std;
int main(void)
{
    int marks;
    cout<<"\nEnter your Marks ";
    cin>>marks;
    if (marks>=80)
        cout<<"Grade is A1";
    else if(marks>=70)
        cout<<" Grade is A";
    else if(marks >=60) cout<<" Grade is B";
    else if(marks>=50) cout<<" Grade is C";
    else
        cout<<" Fail";

    return 0;
}

```



**Q.13** Write a program in C++ using switch case? Or Write a program in C++ using switch case which returns name of weekday depending on the input.

**Ans.**

```

#include<stdio.h>
#include <iostream.h>
using namespace std;
int main(void)
{
    cout<<"\nEnter number of weekday 1 to 7";
    cin>>dow;
    switch(dow)
    {
        case 1:cout<<"Sunday";
        break;
        case 2:cout<<"Monday";

```



```

break;
case 3: cout<<"Tuesday";
break;
case 4: cout<<"Wednesday";
break;
case 5: cout<<"Thursday";
break;
case 6: cout<<"Friday";
break;
case 7: cout<<"Saturday";
break;
default: cout<<"Invalid day number";
return 0;
}

```

Q.14

Write a program in C++ using switch case? OR Write a program in C++ using switch case which returns name of month depending on the input.

Ans.

```

#include<stdio.h>
#include <iostream.h>
using namespace std;
int main(void)
{
cout<<"\nEnter number of month 1 to 12";
cin>>moy;
switch(moy)
{
case 1:cout<<"January";
break;
case 2:cout<<"February";
break;
case 3:cout<<"March";
break;
case 4:cout<<"April";
break;
case 5:cout<<"May";
break;
case 6:cout<<"June";
break;
case 7: cout<<"July";
break;
case 8; cout<<"August";
break;
case 9; cout<<"September";
break;
case 10; cout<<"October";
break;
case 11; cout<<"November";

```



```

break;
case 12; cout<<"December";
break;
default: cout<<"Invalid month number";
return 0;
}

```

**Q.15** What is the purpose of default statement in C++?

**Ans.** A default statement is used when none of the condition is true. No break is needed in the default case

**Q.16** What is loop?

**Ans.** Loop or iteration allows a statement or group of statement to be executed number of times.

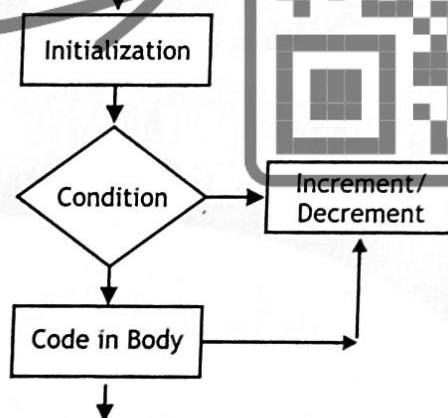
**Q.17** How many types of loops are used in C++?

**Ans.** Three types of loops are used in C++:

- For
- While
- Do-While

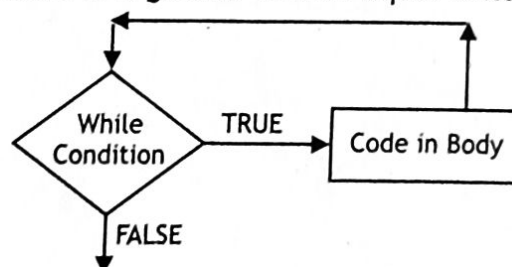
**Q.18** Write the function of "for" loop?

**Ans.** A "For" Loop is used to repeat a specific block of code a known number of times. For example, if we want to check the grade of every student in the class, we loop from 1 to the number of students in the class.



**Q.19** Write the function of "while" loop?

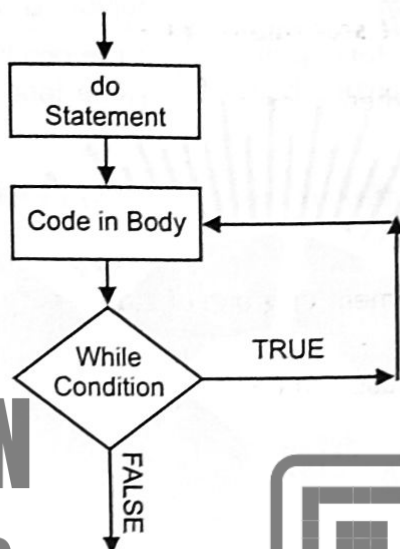
**Ans.** The while loop is used to repeat a section of code an unknown number of times until a specific condition is met. For example, say we want to know how many times a given number can be multiplied by 2 before it is greater than or equal to 20.





**Q.20** Write the function of "do-while" loop?

**Ans.** A do-while loop is a control flow statement that executes a block of code at least once, and then either repeatedly executes the block, or stops executing it, depending on a given boolean condition at the end of the block. do-while loop verifies the condition after the execution of the statements inside the loop.



**Q.21** Differentiate between while and do-while loop?

**Ans.**

While loop	Do-While loop
while ( condition ) { statements; // body of loop }	do { statements; // body of loop. } while (Condition);
In 'while' loop the controlling condition appears at the start of the loop.	In 'do-while' loop the controlling condition appears at the end of the loop.
It is also known as an entry-controlled loop	It is also known as an exit-controlled loop
No semi-colon is used	Semi-colon is used

**Q.22** Can we use while loop in place of for loop?

**Ans.** Yes. A while loop can be used in place of for loop. A for loop is used when the number of iterations are known whereas while loop is generally used when the number of iterations are not predefined.

**Q.23** Why we use block of statement using braces?

**Ans.** Using braces increase the visibility and readability of the code and especially the loop body.

**Q.24** Which data type can be used in switch statement?

**Ans.** A switch statement can use integer and character as data type.

**Q.25** What is the difference between for and while?

**Ans.** Following is the main difference between for and while loop:

<i>for loop</i>	<i>While loop</i>
The structure of for loop is - for(initial condition; number of iterations){//body of the loop }	Structure of while loop is While (condition) {statements;//body}
Iterates for a preset number of times.	Iterates till a condition is met.
Used to obtain the result only when the number of iterations is known.	Used to obtain the result only when the number of iterations is known.
The structure of for loop is - for(initial condition; number of iterations){//body of the loop }	If the condition is not mentioned in the 'while' loop, it results in a compilation error.

**Q.26** Write a program to take number as input and returns whether the number is even or odd.

**Ans.** `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{`  
`int n;`  
`cout << "Enter an integer: ";`  
`cin >> n;`  
`if ( n % 2 == 0)`  
`cout << n << " is even.";`  
`else`  
`cout << n << " is odd.";`  
`return 0;`  
`}`



**Q.27** Write a program to add numbers from 1-20?

**Ans.** `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{`  
`int i,sum=0;`  
`cout << "\n\n Find the first 20 numbers:\n";`  
`cout << "-----\n";`  
`cout << " The numbers are: \n";`  
`for (i = 1; i <= 20; i++)`  
`{`  
`cout << i << " ";`  
`sum=sum+i;`  
`}`  
`cout << "\n The sum of first 20 numbers: "<<sum << endl;`  
`}`



**Q.28** Write a program to print sum of N even numbers?

**Ans.** `#include <iostream>`

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int j, n, sum = 0;
```

```
    // Take input from user.
```

```
    cout << "Print sum of even numbers till : ";
```

```
    cin >> n;
```

```
    for(j = 1; j <= n; j++)
```

```
{
```

```
    // Check for even or not.
```

```
    if((j % 2) == 0)
```

```
{
```

```
        sum += j;
```

```
}
```

```
cout << endl << "Sum of even numbers from 1 to " << n << " is : " << sum;
```

```
return 0;
```

```
}
```

# JOIN

# FOR

# MORE!!!

Q.29

Write a C++ program which takes a six digit number as input and print each digit in a separate line?

Ans.

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num, digit1, digit2, digit3, digit4, digit5, digit6;
```

```
    cout << "Enter a 6 digit number: ";
```

```
    cin >> num;
```

```
    digit6 = num % 10;
```

```
    num = num / 10;
```

```
    digit5 = num % 10;
```

```
    num = num / 10;
```

```
    digit4 = num % 10;
```

```
    num = num / 10;
```

```
    digit3 = num % 10;
```

```
    num = num / 10;
```

```
    digit2 = num % 10;
```

```
    num = num / 10;
```

```
    digit1 = num % 10;
```

```
    cout << digit1 << "\n" << digit2 << "\n" << digit3 << "\n" << digit4 << "\n" << digit5 << "\n" << digit6 << "\n";
```

```
    digit6 << "\n";
```

```
    return 0;
```

```
}
```

Q.30

Write a C++ program which takes character as input and ask for rows and columns. Draw a square box filled with the character?

Ans. #include <iostream>



```

#include <conio.h>
using namespace std;
int main()
{
    int rows, columns, i, j;
    cout<<"Enter the number of rows: ";
    cin>>rows;
    //Takes input from user for rows
    cout<<"Enter the number of columns: ";
    cin>>columns;
    //Takes input from user for columns
    for(i=1; i<=rows; i++){//outer for loop
        for (j=1; j<=columns; j++){//inner for loop
            cout<<"*"; //print star
        }
        cout<<"\n"; //move to next line
    }
    getch();
    return 0;
}

```

**Q.31** Write a C++ program which prints hollow rectangle with \*?

**Ans.**

```

#include <bits/stdc++.h>
using namespace std;
// Function to print hollow rectangle
void print_rectangle(int n, int m)
{
    int i, j;
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= m; j++)
        {
            if (i == 1 || i == n ||
                j == 1 || j == m)
                cout << "*";
            else
                cout << " ";
        }
        cout << endl;
    }
}

// Driver Code
int main()
{
    int rows = 6, columns = 20;
    print_rectangle(rows, columns);
    return 0;
}

```



**Q.32** Write a C++ program which prints half star pyramid with \*?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    int i, j, n;
    cout << "Enter number of rows: ";
    cin >> n;
    for(i = 1; i <= n; i++)
    {
        for(j = 1; j <= i; j++)
        {
            cout << "* ";
        }
        //Ending line after each row
        cout << "\n";
    }
    return 0;
}
```

**JOIN  
FOR  
MORE!!!**

**Q.33** Write a C++ program which prints star pyramid with \*?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    int n, s, i, j;
    cout << "Enter number of rows: ";
    cin >> n;
    for(i = 1; i <= n; i++)
    {
        //for loop for displaying space
        or(s = i; s < n; s++)
        {
            cout << " ";
        }
        //for loop to display star equal to row number
        for(j = 1; j <= (2 * i - 1); j++)
        {
            cout << "*";
        }
        // ending line after each row
        cout << "\n";
    }
    return 0;
}
```





**Q.34** Write a program which takes number as input and checks if it is prime or not?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    bool flag = true;
    cout<<"Enter any number(should be positive integer): ";
    cin>>num;
    for(int i = 2; i <= num / 2; i++) {
        if(num % i == 0) {
            flag = false;
            break;
        }
    }
    if (flag==true)
        cout<<num<<" is a prime number";
    else
        cout<<num<<" is not a prime number";
    return 0;
}
```

**Q.35** Write a C++ to display square of number from 1 to 10?

**Ans.**

```
#include <iostream>
int main()
{
    std::cout<<"1 2 3 4 5 6 7 8 9 10\n1 4 9 16 25 36 49 64 81 100\n";
    getch();
}
```

**Q.36** Write a C++ to take number as input and display its square?

**Ans.**

```
#include <iostream.h>
using namespace std;
int main()
{
    int num,sqr;
    cout << "Enter a number \n";
    cin >> num;

    // Calculate square of a number
    sqr = num * num;
    cout << " Square of a number is " << sqr;
    return 0;
}
```



**Q.37** Write a C++ to take number as input and display its cube?

**Ans.**

```
#include<iostream.h>
using namespace std;
int main()
{
    int num, cube;
    cout << "Enter number \n";
    cin >> num;
    /* Cube calculation. */
    cube = num * num * num;
    cout << "Cube of a number is" << cube;
    return 0;
}
```

**Q.38** Write a C++ program which takes a number and print its table from 1 to 10?

**Ans.**

```
#include<iostream>
using namespace std;
int main()
{
    int num, i, res;
    cout<<"Enter the Number: ";
    cin>>num;
    for(i=1; i<=10; i++)
    {
        res = num*i;
        cout<<num<<" * "<<i<<" = "<<res;
        cout<<endl;
    }

    cout<<endl;
    return 0;
}
```



**Q.39** Write a program to print half pyramid of numbers?

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    int rows;
    cout << "Enter number of rows: ";
    cin >> rows;
    for(int i = 1; i <= rows; ++i)
    {
        for(int j = 1; j <= i; ++j)
        {
            cout << j << " ";
        }
    }
}
```



```

    }
    cout << "\n";
}
return 0;
}

```

**Q.40** Write a program to print half pyramid of same numbers?

**Ans.**

```

#include <iostream>
using namespace std;
int main()
{
    int rows;
    cout << "Enter number of rows: ";
    cin >> rows;
    for(int i = 1; i <= rows; i++)
    {
        for(int j = 1; j <= i; j++)
        {
            cout << i;
        }
        cout << endl;
    }
}

```

**Q.41** Write a C++ program to take salary as input and print designation?

**Ans.**

```

#include<iostream>
using namespace std;
int main()
{
    float basic_salary;
    cout<<"Enter basic salary of Employee : ";
    cin>>basic_salary;

    if (basic_salary<=1500)
    {
        cout<<"Employee Designation is Supervisor ";
    }
    else
    {
        cout<<"Employee Designation is Manager";
    }

    return 0;
}

```



**Q.42** What is the purpose of Jump statement?

**Ans.** Jump statement is used to change the sequence of execution with a program.

**Q.43** Explain some jump statements used in C++?

**Ans.** Following is the description of jump statements used in C++:

**Break:**

It terminates the loop or switch statement and transfers control to the statement immediately following the loop or switch statement.

**Continue:**

It causes the loop to skip the remaining statements of its body and immediately transfer the control to top of the loop.

**Goto:**

It transfers control unconditionally from the "goto" to a labelled statement in the same function.

**Return:**

It terminates the execution of a function and transfers program control to the statement just after the function call statement in the calling function.

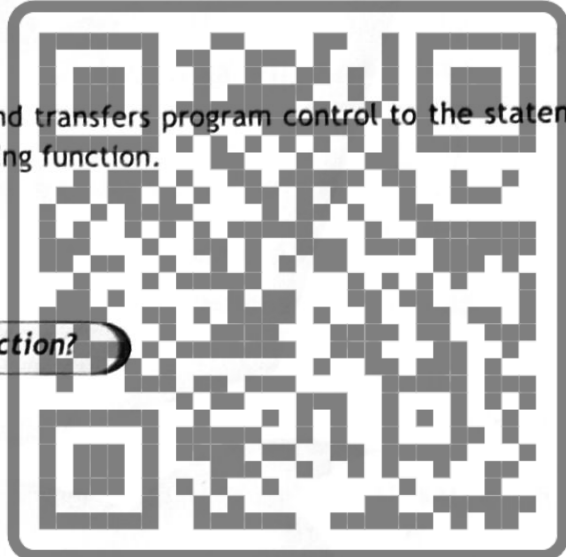
**Exit:**

It is used to terminate C++ program.

**Q.44** Write a C++ which uses "break" jump function?

**Ans.**

```
#include<iostream>
using namespace std;
int main ()
{
    int x=0, y, sum=0;
    cout<<"Enter a number: ";
    cin>>y;
    while(1) {
        x++;
        if (x>y)
            break;
        if(y%x!=0)
            continue;
        sum=sum+x;
    }
    cout<<"\n Sum of factors: "<<sum;
    return 0;
}
```



**Q.45** Write a C++ which uses "goto" jump function?

**Ans.**

```
#include<iostream>
using namespace std;
```

```

int main ()
{
    int x = 10;
    loop: cout<<x<<" "; //loop is a label
    x--;
    if (x<0)
        goto loop;
    cout<<"\n Here is the example of goto !";
    return 0;
}

```

**Q.46** Write a C++ which uses "exit" jump function?

**Ans.**

```

#include<iostream>
#include<cstdlib> //for exit() function
using namespace std;
int main ()
{
    int a;
    cout<<"Enter the value for a: ";
    while(cin>>a) {
        if(a<0) {
            cout<<"This program is going" <<"to terminate!";
            exit(0);
        }
        cout<<"Enter another value for a: ";
    }
    return 0;
}

```

JOIN  
FOR  
MORE!!!



**Q.47** Write a C++ which uses switch statement and display name of color based on value?

**Ans.**

```

#include<iostream.h>
#include<conio.h>
void main()
{
    int code;
    clrscr();
    cout<<"Enter the color code : ";
    cin>>code;
    if(code==1 )
    {
        cout<<"Color is Red";
    }
    else if(code==2)
    {
        cout<<" Color is Green";
    }
    else if(code==3)
    {
        cout<<" Color is White";
    }
}

```





```

    }
    else if(code==4)
    {
        cout<<"Color is Yellow";
    }
    else
    {
        cout<<" No Color code defined";
    }

    cout<<endl;
    cout<<"End of the Program ";
    getch();
}

```

**Q.48** Write a C++ to identify vowel using switch case?

**Ans.**

```

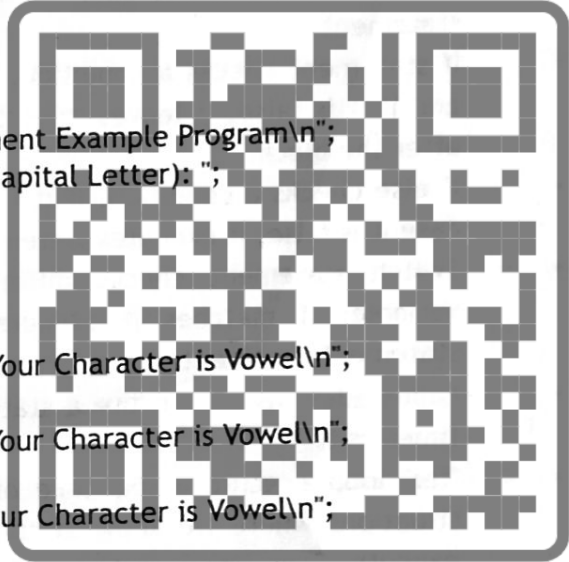
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    char ch;
    cout << "Simple Switch Statement Example Program\n";
    cout << "Enter the Letter (In Capital Letter): ";
    cin >> ch;

    //Switch Case Check
    switch (ch) {
        case 'A': cout << "Your Character Is A. Your Character is Vowel\n";
            break;
        case 'E': cout << "Your Character Is E. Your Character is Vowel\n";
            break;
        case 'I': cout << "Your Character Is I. Your Character is Vowel\n";
            break;
        case 'O': cout << "Your Character Is O. Your Character is Vowel\n";
            break;
        case 'U': cout << "Your Character Is U. Your Character is Vowel\n";
            break;
        default: cout << "Your Character is Not Vowel. Otherwise Not a Capital Letter\n";
            break;
    }

    // Wait For Output Screen
    getch();

    //Main Function return Statement
    return 0;
}

```



**Q.49** Match the following column?

**Ans.**

S.NO	A	S.NO	B	Correct Pair
(a)	If	(I)	Relational operator	(a)(vi)

(b)	Loop	(ii)	Break	(b)(v)
(c)	conditional statement	(iii)	Switch	(c)(i)
(d)	Loops and switch	(iv)	Operator	(d)(ii)
(e)	Do	(v)	Iteration	(e)(vii)
(f)	>>	(vi)	Else	(f)(iv)
(g)	case	(vii)	While	(g)(iii)

## Summary

- C++ has three types of control statements: Selection/Decision Making Structure, Iteration / Loops and Jump.
- C++ has three decision making structures; 'if' statement, 'if-else' statement and 'switch' statement.
- If statement checks a condition, if it is true the statements in if block are executed and if it is false, it leaves the statements in if block and starts executing statements after the block.
- If else checks a condition, if it is true the statements in if block are executed and in case it is false, it executes statements in 'else' block.
- Switch statement checks different constants after case statement with switch variable; if matches it executes statements after it otherwise goes to default statement if present.
- Loops allow us to execute a statement or a group of statements several numbers of times.
- "for" loop execute a sequence of statements multiple times. And is usually used in situations where we know at the start of loop that how many times loop body will execute. Condition is tested at the start of loop.
- Like for loop "while" loop also repeats a statement or group of statements several numbers of times while a given condition is true. It tests the condition at start of loop and is usually used in situations where we do not know at the start of loop that how many times loop block will execute.
- "do while" loop is similar to "while" loop, except that it tests the condition at the end of the loop body. So its statements block is executed at least one time.
- If a loop exists in the body of another loop then it is called nested loop.
- A "break" statement terminates the loop or switch statement and transfers control to the statement immediately following the loop or switch statement.
- A "continue" statement causes the loop to skip the remaining statements of its body and immediately transfers control to the top of the loop.
- A "goto" statement jumps or transfers control unconditionally from the "goto" to a labeled statement in the same function.
- A "return" statement terminates the execution of a function and transfers program control to the statement just after the function call statement in the calling function.
- The `exit()` is used to terminate a C++ program.

# Solution of Textbook Exercise

## A. Encircle the correct answer:

Ans. See "Multiple Choice Question (MCQS)" Q1-10

## B. Respond the following:

1. What is the purpose of "default" statement in C++?

Ans. See "Short/Detailed Questions and Answers" – Q.15

2. Can we use "while" loop in place of "for" loop, if yes then how?

Ans. See "Short/Detailed Questions and Answers" – Q.22

3. What is the main difference between while and do while loops?

Ans. See "Short/Detailed Questions and Answers" – Q.21

4. Write the function of for loop.

Ans. See "Short/Detailed Questions and Answers" – Q.18

5. Why we make block of statements using braces?

Ans. See "Short/Detailed Questions and Answers" – Q.23

6. Which data type variables can be used in "switch" statement?

Ans. See "Short/Detailed Questions and Answers" – Q.24

7. What is the purpose of jump statements?

Ans. See "Short/Detailed Questions and Answers" – Q.42

8. Write the purpose of following statements:

a. else if      b. switch      c. goto d. exit

Ans. See "Short/Detailed Questions and Answers" – Q.43

9. Match the following table?

S.NO	A	S.NO	B
(a)	If	(i)	Relational operator
(b)	Loop	(ii)	Break
(c)	conditional statement	(iii)	Switch
(d)	Loops and switch	(iv)	Operator
(e)	Do	(v)	Iteration
(f)	>>	(vi)	Else
(g)	case	(vii)	While

Ans. See "Short/Detailed Questions and Answers" – Q.49



## Lab Activity

**1.** Write a program that takes a number as input and print whether it is odd or even.

**Ans.** See "Short/Detailed Questions and Answers"– Q.26

**2.** Write a program to add numbers from 1 to 20.

**Ans.** See "Short/Detailed Questions and Answers"– Q.27

**3.** Write a program that take month number as input (from 1 to 12) and print number days in that month. If wrong number is given then show error message.

**Ans.** See "Short/Detailed Questions and Answers"– Q.14

**4.** Input a number up to six digits and show each digit in separate line.

**Ans.** See "Short/Detailed Questions and Answers"– Q.29

**5.** Take input a character, number of rows and number of columns. Draw a square box filled with that character with given number of rows and columns.

**Ans.** See "Short/Detailed Questions and Answers"– Q.30

**6.** Write a program that generate the following outputs

*	1	1
**	12	22
***	123	333
****	1234	4444
*****	12345	55555

**Ans.** See "Short/Detailed Questions and Answers"– Q.32,Q.39,Q.40

**7.** Write a program that takes a number as input and print whether it is prime or not.

**Ans.** See "Short/Detailed Questions and Answers"– Q.34

**8.** Take salary as input and on its basis show different levels of designations in an organization like manager, supervisor, worker etc.

**Ans.** See "Short/Detailed Questions and Answers"– Q.41

**9.** Write a program that prints square of numbers from 1 to 10.

**Ans.** See "Short/Detailed Questions and Answers"– Q.35

**10.** Take a number and print its table from 1 to 10 using while loop according to the following format.

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

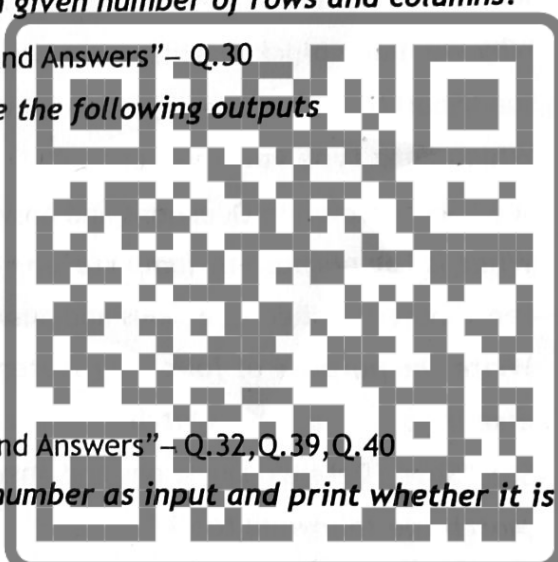
$$2 \times 3 = 6$$

.

.

$$2 \times 10 = 20$$

**Ans.** See "Short/Detailed Questions and Answers"– Q.38





# UNIT 05

## FUNCTIONS

### Function Prototype

return type      function name      parameters (arguments)

HEADER { int heading ( void ) ← NO semicolon

BODY { //statements  
return 0;

### C++ Vector Functions



www.educha.com

### Function Definition in C

```
int add(int, int);

int main()
{
    int m = 20, n = 30, sum;
    sum = add(m, n);
    printf("sum is %d", sum);
}

int add(int a, int b)
{
    return (a + b);
}
```

This is the way how you define a function.

NOTE: it is important to mention both data type and name of parameters

63

C Programming





# Multiple Choice Questions (M.C.Qs)

**Choose the right answer:**

1. The functions that are defined by the programmer are called:
  - (a) Built-In function
  - (b) User-defined-function
  - (c) Sub function
  - (d) Function
2. A programmer creates a function for a particular task and the programmer wants to include that function in program. Which extension is required to save that function?
  - (a) .obj
  - (b) .j
  - (c) .cpp
  - (d) .exe
3. In C++, int main( ) returns which data type value by default?
  - (a) float
  - (b) integer
  - (c) character
  - (d) double
4. The parameters specified in the function header are called:
  - (a) formal parameters
  - (b) actual parameters
  - (c) default parameters
  - (d) command line parameters
5. The word "prototype" means:
  - (a) Declaration
  - (b) Calling
  - (c) Definition
  - (d) Both a & b
6. The function prototype consists of:
  - (a) Name of function:
  - (b) the parameters are passed to the function
  - (c) The value return from function
  - (d) All of the above
7. All variables declared in function definition are called.
  - (a) Local variable
  - (b) Instance variable
  - (c) Global variable
  - (d) Static variable
8. Which are not the built-In function?
  - (a) sqrt()
  - (b) time()
  - (c) exp0
  - (d) sin()
9. A set of statements used to perform specific task is called as
  - (a) comment
  - (b) function
  - (c) statement
  - (d) library
10. "When complicated programs are broken down into smaller modules it is termed as
  - (a) small function
  - (b) micro function
  - (c) subgroup
  - (d) subprogram
11. Every program has atleast \_\_\_\_\_ main() function
  - (a) one
  - (b) two
  - (c) three
  - (d) four
12. When a program starts main() function is called for
  - (a) execution
  - (b) debugging
  - (c) jump
  - (d) compilation
13. There are \_\_\_\_\_ types of function
  - (a) one
  - (b) two
  - (c) three
  - (d) four

14. \_\_\_\_\_ functions are part of every high-level program  
 (a) user-defined (b) post-defined (c) pre-defined (d) None of these
15. Pre-defined functions are also called as  
 (a) stationary (b) studio (c) standard (d) library
16. Pre-defined functions are declared in the  
 (a) body (b) loop (c) header file (d) end
17. Functions contains which symbol  
 (a) {} (b) "" (c) [] (d) []
18. Definition of pre-defined functions can be found in  
 (a) cstdlib (b) cstdio (c) conio (d) include
19. Customized functions created by user are called as  
 (a) user-defined (b) post-defined (c) pre-defined (d) None of these
20. A user defined function has \_\_\_\_\_ parts  
 (a) one (b) two (c) three (d) four
21. A function without its code block is called as  
 (a) Prototype (b) Data type (c) Initialization (d) definition
22. Following is the correct syntax of function declaration  
 (a) Int Office (int value1, int value2) (b) Char Office (int value1, int value2)  
 (c) Int Office (Char value1, int value2) (d) Int Office (int value1, Char value2)
23. Function prototype end with  
 (a) "" (b) {} (c) // (d) ;
24. First part of function prototype is called as  
 (a) Return datatype (b) Name (c) Parameter (d) Terminator
25. Second part of function prototype is called as  
 (a) Return datatype (b) Name (c) Parameter (d) Terminator
26. Third part of function prototype is called as  
 (a) Return datatype (b) Name (c) Parameter (d) Terminator
27. Fourth part of function prototype is called as  
 (a) Return datatype (b) Name (c) Parameter (d) Terminator
28. If no value is returned by the function then \_\_\_\_\_ keyword is used  
 (a) void (b) main (c) int (d) null
29. Parameters are also called as  
 (a) operator (b) statement (c) library (d) arguments
30. Function body is enclosed in  
 (a) {} (b) {} (c) [] (d) <>



- 31.** Which of the following function declaration using default arguments is incorrect?  
 (a) `int foor(int x, int y =5, int z=10)` (b) `int foo(int x=5, int y =10, int z)`  
 (c) `int foo(int x=5, int y, int z=10)` (d) All are correct
- 32.** Using a function in program requires  
 (a) initialization function (b) calling function  
 (c) declaring function (d) assigning function
- 33.** Passing actual values to function as arguments with function call statement is called as  
 (a) initial parameters (b) actual parameters  
 (c) standard parameters (d) formal parameters
- 34.** Values received in variables of the header of function definition are called as  
 (a) initial parameters (b) actual parameters  
 (c) standard parameters (d) formal parameters
- 35.** In C++ which keyword is used to return value  
 (a) return (b) void (c) assign (d) All of these
- 36.** Which is the correct form of user-defined function  
 (a) `void fog (void)` (b) `void fog (int,float,char)`  
 (c) `int,float,char fog (void )` (d) All of these
- 37.** Which is the incorrect form of user-defined function  
 (a) `int,float,char void (void)` (b) `void tre (int,float,char)`  
 (c) `int,float,char tre (void )` (d) None of the above
- 38.** Variables which are defined within function are called as  
 (a) Global (b) standard (c) local (d) specific
- 39.** Variable declared in the header are also called as  
 (a) local (b) standard (c) global (d) specific
- 40.** A local variable is only accessible within \_\_\_\_ part of program  
 (a) initial (b) last (c) specific (d) any
- 41.** Variables which are defined outside the main function are called as  
 (a) Global (b) standard (c) local (d) specific

## Answers

1. (b)	2. (c)	3. (b)	4. (a)	5. (a)	6. (d)	7. (a)	8. (c)
9. (b)	10. (d)	11. (a)	12. (a)	13. (b)	14. (c)	15. (d)	16. (c)
17. (a)	18. (a)	19. (a)	20. (d)	21. (a)	22. (a)	23. (d)	24. (a)
25. (b)	26. (c)	27. (d)	28. (a)	29. (d)	30. (a)	31. (a)	32. (b)
33. (b)	34. (d)	35. (a)	36. (d)	37. (a)	38. (c)	39. (a)	40. (c)
41. (a)							

# Short & Detailed Answer Questions

**Q1** What is a function?

**Ans.** A set of statements written to perform specific task and having a unique name is called a function.

**Q.2** Define the term Subprogram?

**Ans.** When a program is broken down into small modules, each module is referred to as a subprogram. In C++ a subprogram is a function.

**Q.3** How many types of functions are used in C++?

**Ans.** There are two types of functions in C++:

- Pre-defined function
- User defined function

**Q.4** What are pre-defined functions?

**Ans.** Functions which are already available in C++ library and does not require any definition or declaration are termed as pre-defined functions. These functions are general purpose functions.

**Q.5** What is user defined function?

**Ans.** Functions which are created by the user and requires proper definition and declaration are termed as user defined functions. These functions are used for specific purpose.

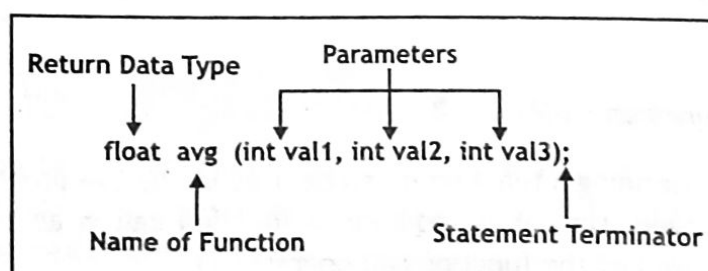
**Q.6** How many segments are there in a user defined function? OR Which are the essential components of user defined functions?

**Ans.** A user defined function is a function which is created by user and performs specific task. It has two major parts:

- Function Declaration
- Function Definition

**Q.7** Explain the term function declaration?

**Ans.** A function without code block is called as function declaration. It is also called as prototype. A function is always declared before the main() function.





**Q.8** Discuss various components of function declaration?

**Ans.** Function declaration has four main components which are:

**Function name:**

It is a unique name and must be meaningful.

**Return Data Type:**

It shows the data type returned by the function.

**Parameters:**

Parameters define the list of data types allowed within the function argument.

**Terminator:**

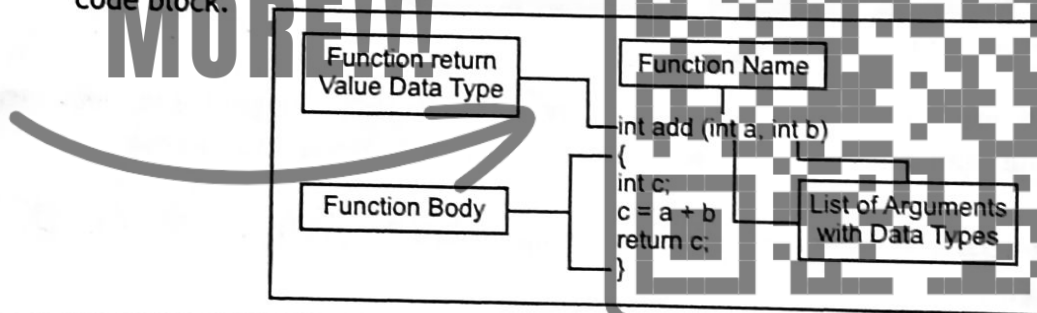
Every function declaration ends with a semi colon which is called as function terminator.

**Q.9** What is the use of keyword void in function?

**Ans.** When used as a function return type, the void keyword specifies that the function does not return a value. When used for a function's parameter list, void specifies that the function takes no parameters. Hence void is a keyword which is used to show null value within a function.

**Q.10** What is meant by function definition?

**Ans.** A function definition is the actual body of the function. It consists of a function header and a code block.



**Q.11** How many parts does a function header has?

**Ans.** A function header has three parts:

- Return value data type
- Function name
- List of arguments

**Q.12** How to write a function body?

**Ans.** A function body can be written before or after the main function, but it should be enclosed within braces { }.

**Q.13** Define the term function call?

**Ans.** After declaring and defining a function it can be used in any C++ program and it is necessary to invoke a function every time it is required. A function call is an expression containing the function name followed by the function call operator, ( )



**Q.14** What is argument?

**Ans.** Arguments are also called as parameters. It is a part of data passed to the function.

**Q.15** Define actual parameters?

**Ans.** When actual values to function as arguments are passed it is termed as actual parameter.

**Q.16** Define formal parameters?

**Ans.** Value which are received in variable of the header of the function definition are called as formal parameters.

**Q.17** Differentiate between function declaration and function definition?

**Ans.**

Function Declaration	Function Definition
It is a prototype	It is the actual function
It specifies function name, data type and arguments without function body	It specifies function name, data type and arguments with function body
It defines the structure of function	It defines the execution of function
Determines the value stored in variable, function	Specifies the name and type of variable, function

**Q.18** What is the use of return keyword?

**Ans.** In C++, a return keyword is used as return function and allows to return a value.

**Q.19** Differentiate between function definition and function call?

**Ans.**

Function Definition	Function Call
It is to invoke the code by function name	It is the actual function
It has function name followed by parameter list	It specifies function name, data type and arguments with function body
It calls a function for execution	It defines the execution of function
Syntax: Data_type function_name (parameter list) { Statements; }	Syntax: Variable_name = function name (parameter list);

**Q.20** Why do we use function header?

**Ans.** The header includes the name of the function and tells the compiler what type of data it expects to receive (the parameters) and the type of data it will return (return value type) to the calling function or program.

**Q.21 Differentiate between passing arguments and returning value to the function?****Ans.**

Passing Arguments	Returning value
They are written after function name	They are written before function name
It is the input structure	It is the output structure
It is separated by comma	It is separated by /
It is the input given to the function	It is the output from the function

**Q.22 What are the different ways to use user defined function?****Ans.** There are four ways to use user-defined functions in C++:**Void function name (Void):**

It is no return value and no passing argument

**Void function name (int,float,char):**

It has no return value but only has passing arguments.

**Int/float/char function name (void):**

It has return value but no passing argument

**Int/float/char function name (int,float,char):**

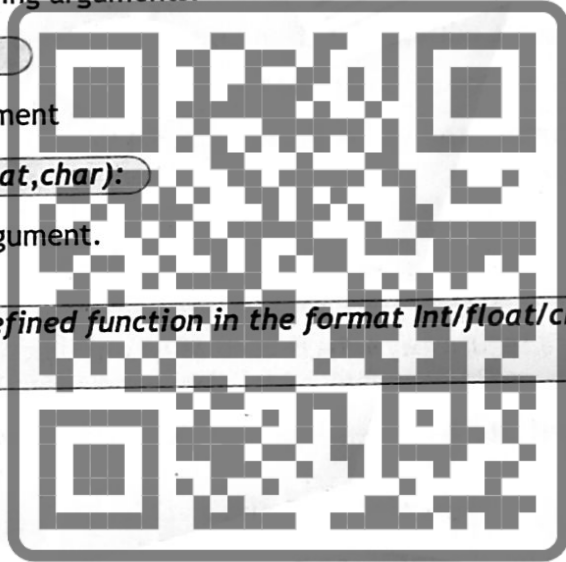
It has both return value and passing argument.

**Q.23****Write a C++ program and use user-defined function in the format Int/float/char function name (int,float,char)?****Ans.**

```

#include<iostream.h>
Using namespace std;
float average(int x, int y)
Int main (void)
{
    Int x,y;
    Average (x,y);
Return 0;
}
Float average (int x, int y)
{
    Cout << "\n \t Enter the value of x.....:";
    Cin >> x;
    Cout << "\n \t Enter the value of y.....:";
    Cin >> y;
    Avg = (x+y)/2;
    Cout << "\n The average of two numbers is ...." << avg;
    Return avg;
}

```



**Q.24** Write a C++ program and use user-defined function in the format void function name (int, float, char) ?

**Ans.**

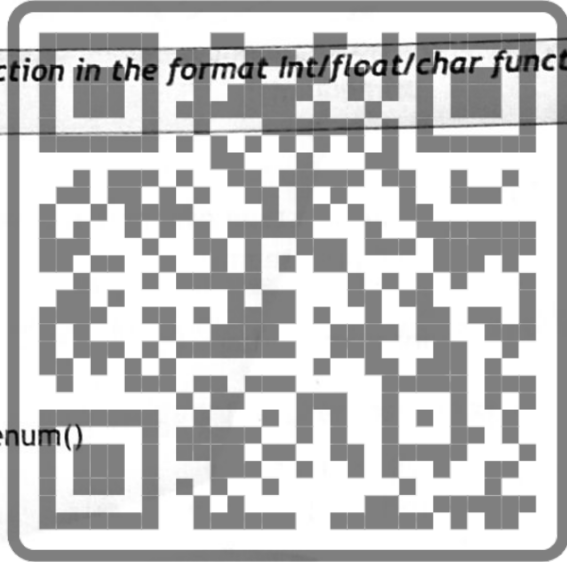
```
#include <iostream>
using namespace std;
void duplicate (int& a, int& b, int& c)
{
    a*=2;
    b*=2;
    c*=2;
}
int main ()
{
    int x=1, y=3, z=7;
    duplicate (x, y, z);
    cout << "x=" << x << ", y=" << y << ", z=" << z;
    return 0;
}
```

**Q.25** Write a C++ program and use user-defined function in the format int/float/char function name (void) ?

**Ans.**

```
#include <iostream>
using namespace std;
int primenum();
int main()
{
    int num, i, flag = 0;
    // No argument is passed to primenum()
    num = primenum();
    for (i = 2; i <= num/2; ++i)
    {
        if (num%i == 0)
        {
            flag = 1;
            break;
        }
    }

    if (flag == 1)
    {
        cout<<num<<" is not a prime number.";
    }
    else
    {
        cout<<num<<" is a prime number.";
    }
    return 0;
}
```



```

    }
    // Return type of function is int
    int primenum()
    {
        int n;

        printf("Enter a positive integer to check: ");
        cin >> n;
        return n;
    }

```

**Q.26** Write a C++ program and use user-defined function in the format void function name (void)?

**Ans.** # include <iostream>  
using namespace std;  
void primenum();  
int main()  
{

```

    // No argument is passed to primenum()
    primenum();
    return 0;
}

```

```

    // Return type of function is void because value is not returned.
    void primenum()
    {

```

```

        int num, i, flag = 0;

```

```

        cout << "Enter a positive integer enter to check: ";
        cin >> num;
        for(i = 2; i <= num/2; ++i)
        {

```

```

            if(num % i == 0)

```

```

        {
            flag = 1;
            break;
        }
    }

```

```

    if (flag == 1)
    {

```

```

        cout << num << " is not a prime number.";
    }

```

```

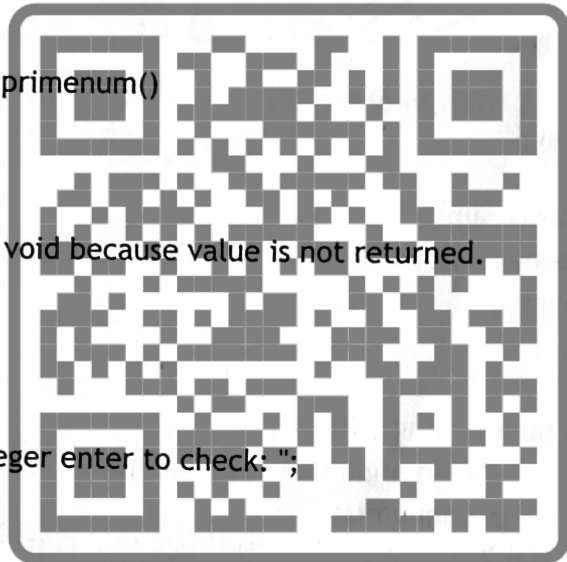
    else
    {

```

```

        cout << num << " is a prime number.";
    }
}

```



**Q.27 Differentiate between predefine and user define functions?**

Ans.

Pre-define	User Define
They are library functions	These are created by programmers
No modification is allowed	Programmer can modify it
No need of function definition	Function declaration and definition both are required
Can be used globally	These functions are used for customized needs

**Q.28 Define local variables?**

Ans.

Variable which are declared inside of any function are called as local variable. These are only accessible within a specific part of program.

**Q.29 Define global variables?**

Ans.

Variable which are declared outside of the main function are called as global variable. Values for global variables can be shared across different functions.

**Q.30 Differentiate between local and global variable?**

Ans.

Local Variable	Global Variable
Defined inside the body of function	Defined outside the body of function
Only has specific access	Can be used across different function
It can be used during specific part of program	It remains available throughout execution
If it is not initialized, a garbage value is stored	If it is not initialized zero is stored as default.

**Q.31 Find error and rectify following C++ programs?**

```
1. int outer, innermost, s;
   for(outer=5; outer>=1; outer--)
   {
       for(s=1 s<=5-outer s++)
       cout<<" ";
       for (innermost=1 innermost<=outer innermost++)
       {
           cout>>"*";
       }
       cout<<endl;
   }
}
```

Ans.

```
int outer, innermost, s;
for(outer=5; outer>=1; outer--)
{
    for(s=1; s<=5-outer; s++)
```



```

        cout<<" ";
        for (innermost=1; innermost<=outer; innermost++)
        {
            cout<<"*";
        }
        cout<<endl;
    }
}

```

**2.** `#include<iostream.h>`  
`void main()`  
`{`  
     `int i=10, j=5;`  
     `int modResult=0;`  
     `int divResult=0;`  
     `modResult = i%j;`  
     `cout<<modResult<<" ";`  
     `divResult = i/modResult;`  
     `cout<<divResult;`  
`}`

**Ans.** `#include<iostream.h>`  
`void main()`  
`{`  
     `int i=10, j=5;`  
     `int modResult=1;`  
     `int divResult=1;`  
     `modResult = i%j;`  
     `cout<<modResult<<" ";`  
     `divResult = i/modResult;`  
     `cout<<divResult;`  
`}`

**3.** `#include <iostream>`  
`using namespace std;`  
`int main()`  
`{`  
     `int i, rows, j, k=1;`  
     `cout<<"Enter the number of rows: ";`  
     `cin<<rows;`  
     `for(i=1; i<=rows; ++i) {`  
         `for(j=1; j<=i; j++) {`  
             `cout<<k<<"\t";`  
             `k++;`  
         `}`  
         `cout<<"\n";`  
`}`  
  
`return 0;`  
`}`



**Ans.**

```
#include <iostream>
using namespace std;
int main()
{
    int i, rows, j, k=1;
    cout<<"Enter the number of rows: ";
    cin>>rows;
    for(i=1; i<=rows; i++) {
        for(j=1; j<=i; j++) {
            cout<<k<<"\t";
            k++;
        }
        cout<<"\n";
    }
    return 0;
}
```

**4.**

```
#include<iostream>
void Table(void);
void Table(void)
{
    int m,n;
    cout << "\n The value of m.....";
    cin >> m;
    for(n=1; n<=10; ++n)
    {
        cout << "\t " << m << "*" << n << " = " << m * n << "\n";
    }
}

Void(main)void;
{
    Table();
}
```



**Ans.**

```
#include<iostream>
using namespace std;
int table(int n);
int main()
{
    int m=0;
    cout << "\n The value of m.....";
    cin >> m;
    table(m);
}

int table(int m)
{
    int n=1;
    cout<<"Table of " <<m << " is: " <<endl;
    for(int i=1; i<=10; i++)
```



```

{
    n=m*i;
    cout<<m <<" * " <<i <<" = " <<n <<endl;
}
return 0;
}

```

**Q.32** List some built in functions with example.

**Ans** strlen(): Used to see string length  
 clrscr(): Used to clear screen  
 Getch: And echo function  
 Pow(); used to find x to the power y  
 Sqrt(); used to find square root of the number  
 Following is the usage of all these five functions:

**a.** #include<iostream.h>  
 #include<string.h>  
 #include<conio.h>  
 void main()  
 {  
     clrscr();  
     char str[80];  
     cout<<"Enter any string (line):\n";  
     cin.getline(str, 80);  
     int len = strlen(str);  
     cout<<"\nLength of the string is: "<<len;  
     getch();  
 }

**b.** #include <iostream>  
 #include <cmath>  
 using namespace std;

```

int main() {
    cout << "Square root of 49 = ";
    cout << sqrt(49);

    return 0;
}

```

**c.** #include <iostream>  
 #include <cmath>  
 using namespace std;

```

int main(){
    /* Calling the built-in function
    * pow(x, y) which is x to the power y
    * We are directly calling this function
    */

    cout<<pow(2,5);
    return 0;
}

```



**Q.33 Write down advantages of user defined function?**

**Ans.** User-defined functions help to break a large program into small segments which makes program easy to understand, maintain and debug. If repeated code occurs in a program, Function can be used to include those codes and execute when needed by calling that function.

**Q.34 Write down advantages of built-in function?**

**Ans.** Built-in functions or library functions are predefined in standard libraries which makes it easier to use and repeat. Predefined functions does not require any declaration or definition.

**Q.35 Write down disadvantages of user defined function? Or Write some drawbacks of user defined functions**

**Ans.** User-defined functions needs declaration and definition in a standardized manner which makes it more complex. These functions cannot be used across programs and requires high end programming skills.

**Q.36 Write a C++ to print 0,5,10,15,20,25,30,35?**

**Ans.**

```
#include <iostream>
using namespace std;
int main()
{   int num = 0;
    while (num < 40)
    {   num = num + 5;
        cout << num << endl;
    }
}
```



**Q.37 Write a C++ to check if the number is even or odd?**

**Ans.**

```
#include <iostream>
#include <conio.h>
using namespace std;
int check_Oddeven(int); //function prototype
int main()
{
    int num;
    cout << "Enter a number" << endl;
    cin >> num; //get input from user
    check_Oddeven(num); //calling the function
    getch();
    return 0;
}

//create function
int check_Oddeven(int num){ //function definition
if(num%2==0)
```



```

        cout<<num<<"is an even";
    else
        cout<<num<<"is an odd";
}

```

**Q.38 Write a C++ to convert kilogram into gram?**

**Ans.** #include <iostream>  
using namespace std;  
double convert(double\* kilograms);  
//main method  
int main()  
{  
double kilograms;  
//prompt user to enter kilograms  
cout<<"Enter kilograms: ";  
cin>>kilograms;  
//show result  
cout<<kilograms<<" kilogram(s) = "<<convert(&kilograms)<<" gram(s).\n";  
system("pause");  
//exit program  
return 0;  
}  
//function convert to convert kilogram to gram by passing pointer as argument to the function  
double convert(double\* kilograms){  
return \*kilograms\*1000;  
}



**Q.39 Write a C++ program to convert celsius into fahrenheit?**

**Ans.** /\*  
\* C++ program to perform celsius to fahrenheit conversion  
\*/  
#include<iostream>  
using namespace std;  
int main()  
{  
float fahrenheit, celsius;  
cout << "Enter the temperature in Celsius : ";  
cin >> celsius;  
fahrenheit = (celsius \* 9.0) / 5.0 + 32;  
cout << "The temperature in Celsius is : " << celsius << endl;  
cout << "The temperature in Fahrenheit is : " << fahrenheit << endl;  
return 0;  
}





Q.40

Create a function that takes length and height as arguments and print a box of stars accordingly.

Ans.

```
/* program to print solid star square using function */
#include <iostream>
using namespace std;
void sqpatt(int,int); //function prototype
int main() //function main begins program execution
{
    int i,j;
    sqpatt(i,j); //function call
    return 0;
} // end main
void sqpatt(int,int) //called function
{
    int i,j,size;
    cout << "Please enter the size" << endl;
    cin >> size; //Takes input from the user for size
    for(i=1; i<=size; i++){
        for(j=1; j<=size; j++){
            cout << "*";
        }
        cout << "\n";
    }
}
```

**JOIN  
FOR  
MORE!!!**



**Q.41** Create a function that creates factorial of a given number.

Ans.

```
#include <iostream>
using namespace std;
int factori(int n);
int main()
{
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;
    cout << "Factorial of " << n << " = " << factori(n);
    return 0;
}
int factori(int n)
{
    if(n > 1)
        return n * factori(n - 1);
    else
        return 1;
}
```



**Q.42** Create a function to find area by  $a=lb$ .

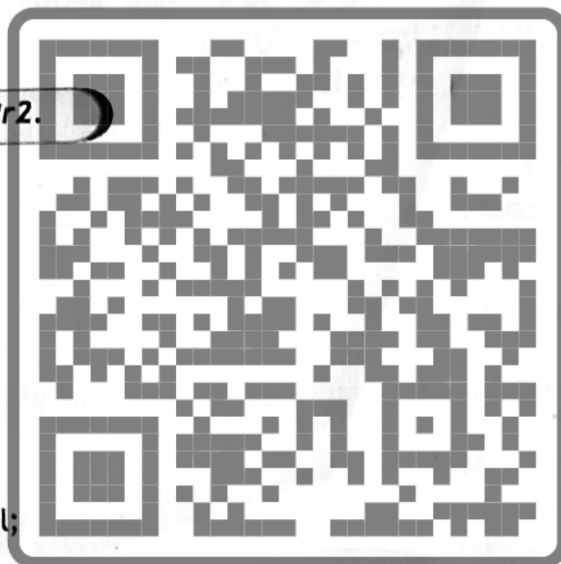
**Ans.**

```
#include<iostream>
using namespace std;
float areaOfSquare(float, float);
int main()
{
    float len, bre, area;
    cout<<"Enter the Length of Square: ";
    cin>>len;
    area = areaOfSquare(len, bre);
    cout<<"\nArea = "<<area;
    cout<<endl;
    return 0;
}
float areaOfSquare(float len, float bre)
{
    return (len*bre);
}
```

**Q.43** Create a function to find area by  $A=PI*r^2$ .

**Ans.**

```
#include <iostream>
#define PI 3.14159
using namespace std;
float AreaOfCircle(float radius);
float AreaWithDiameter(float diameter);
int main() {
    float radius, diameter, circleArea;
    char choice='0';
    cout<<"\n\t\t\tFind Area Of Circle:"<<endl;
    for (;choice!='1'&&choice!='2';) {
        cout<<choice;
        if(choice!='1'&&choice!='2')
            cout<<"\n\t\t\tEnter a VALID Option ";
    }
    if(choice=='1') {
        cout <<radius;
        circleArea=AreaOfCircle(radius);
    } else if(choice=='2') {
        cout <<diameter;
        circleArea=AreaWithDiameter(diameter);
    }
    cout<<" "<<circleArea<<endl;
    return 0;
}
float AreaOfCircle(float radius) {
    return (PI*(radius*radius));
}
```



```

}
float AreaWithDiameter(float diameter) {
return (AreaOfCircle(diameter/2));
}

```

**Q.44** Create a function to find Pythagorean theorem.

**Ans.**

```

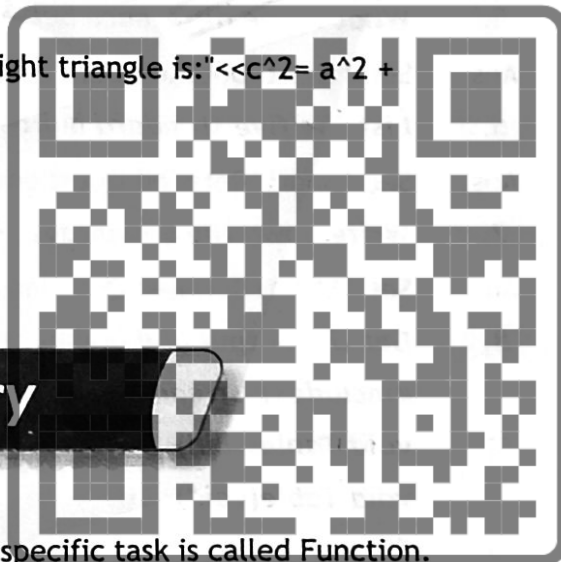
#include <iostream>
using namespace std;
void char RTriangle ( a, b );
int a, b, c, side1, side 2;
int main ( )
{
    cout<< "Please enter side1:" <<endl;
    cin >>side1;
    cout <<"Please enter side 2:" <<endl;
    cin >> side2;
    c=sqrt( a*a + b*b)
    {
        cout<<"The Hypotenuse of of the right triangle is:"<<c^2= a^2 +
        b^2.<<endl;
    }
    return 0;
}

```

**JOIN  
FOR  
MORE!!!**

## Summary

- A group of statements written to perform specific task is called Function.
- Function in C++ helps the programmer to manage the code of the large program.
- Functions are divided into three sections:
  - Function declaration
  - Function definition
  - Function calling
- Function declaration tells the compiler about the function name, return types and parameters types.
- Function call is to invoke the code of function by its name. Functions are divided into two categories.
  - User - defined-function
  - Pre-defined - function
- A programmer can write his/her own function which is called User - defined function.
- In C++, Pre-defined - function are already declared in header files.



# Solution of Textbook Exercise

## A. Encircle the correct answer:

Ans. See "Multiple Choice Question (MCQS)" Q1-10

## B. Respond the following:

1. Differentiate between function declaration and function definition.?

Ans. See "Short/Detailed Questions and Answers" – Q.17

2. What is the purpose of keyword "void" in function?

Ans. See "Short/Detailed Questions and Answers" – Q.9

3. Why we use header files?

Ans. See "Short/Detailed Questions and Answers" – Q.20

4. Differentiate between passing argument and return the value from Function?

Ans. See "Short/Detailed Questions and Answers" – Q.21

5. What is the difference between external variables and function local variables?

Ans. See "Short/Detailed Questions and Answers" – Q.30

6. List the five standard built-in functions with examples?

Ans. See "Short/Detailed Questions and Answers" – Q.32

7. Write down the advantages of User - define functions in C++?

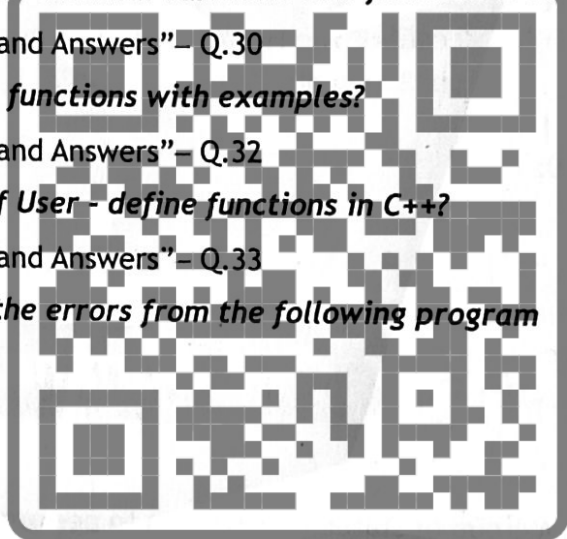
Ans. See "Short/Detailed Questions and Answers" – Q.33

8. Get the output and highlight the errors from the following program

```
#include<iostream>
void Table(void);
void Table(void)
{
    int m,n;
    cout << "\n The value of m.....";
    cin >> m;
    for(n=1; n<=10; ++n)
    {
        cout << "\t " << m << "*" << n << "=" << m * n << "\n";
    }
}

Void(main)void;
{
    Table();
}
```

Ans. See "Short/Detailed Questions and Answers" – Q.31



## Lab Activity

1. Write a program to print following series by using for loop.  
0,5,10,15,20,25

Apply the technique no return value no pass parameter

Ans. See "Short/Detailed Questions and Answers" – Q.36

2. Write a program to take input from the keyboard and check whether given number is Even or Odd. Apply the technique (return value and pass parameters) in program.

Ans. See "Short/Detailed Questions and Answers" – Q.37

3. Write a program to convert kilogram in grams using function. The function should take value in kilogram as parameter and should return value in grams..

Ans. See "Short/Detailed Questions and Answers" – Q.38

4. Create a function that takes length and height as arguments and print a box of stars accordingly. e.g. length =10, height = 4

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Apply the technique (return value but not pass parameters) in program.

Ans. See "Short/Detailed Questions and Answers" – Q.40

5. Write a function that returns factorial of a given number.

Ans. See "Short/Detailed Questions and Answers" – Q.41

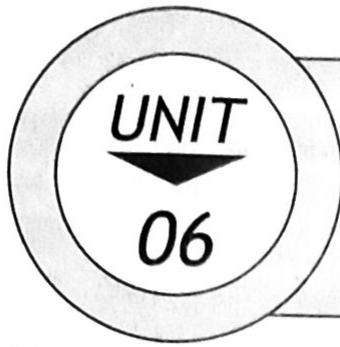
6. Develop programs for manipulating the following formulas in form of function.

Title	Formula	Description
Area of rectangle	$A=lb$	area = length x width
Area of circle	$A= \pi r^2$	area = pi x radius x radius (pi = 3.14)
Pythagorean Theorem	$C^2=a^2+b^2$	*****

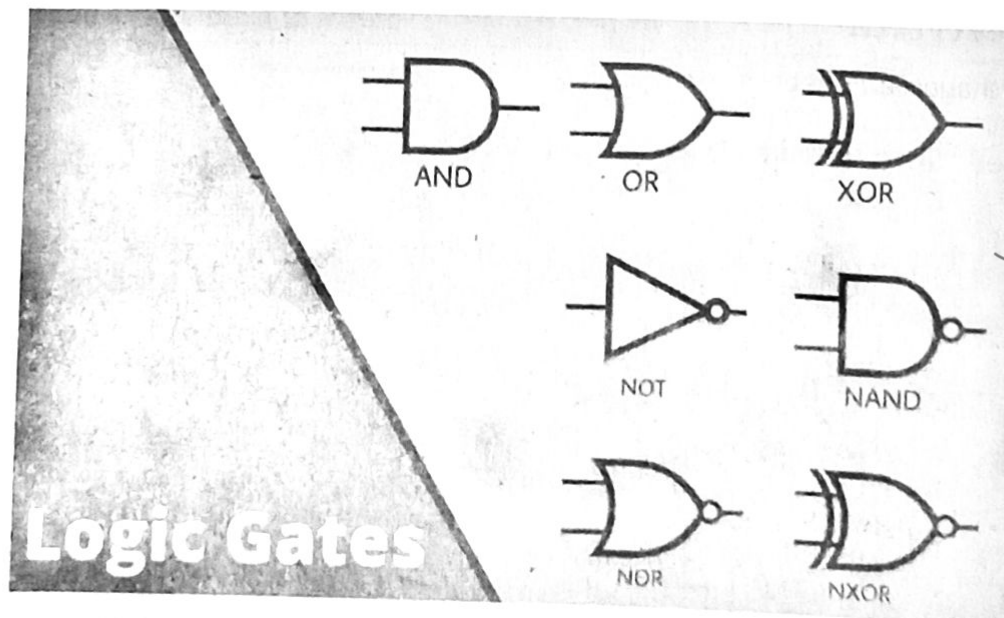
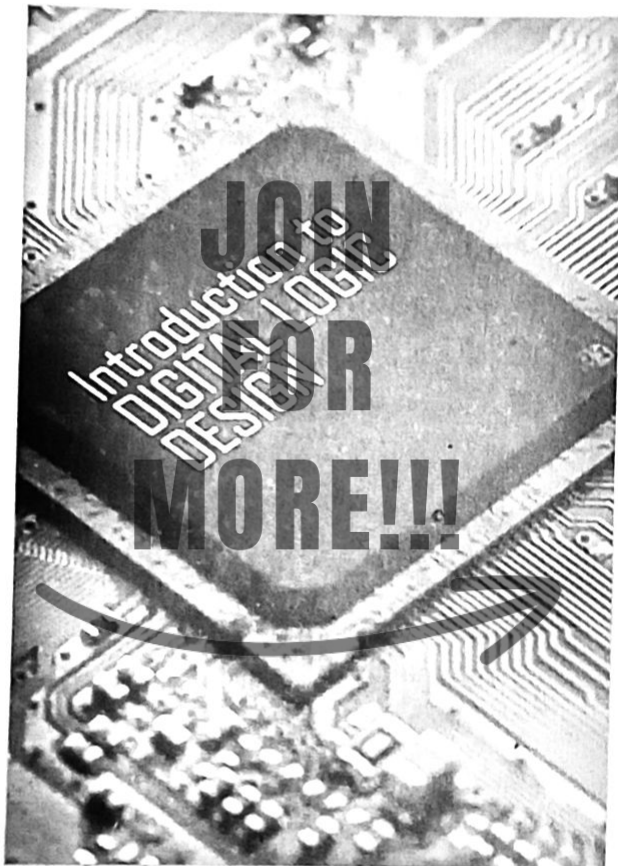
Ans. See "Short/Detailed Questions and Answers" – Q.42 , Q.43 , Q.44





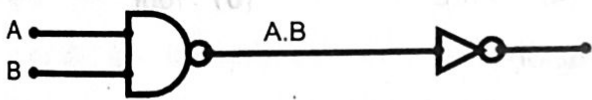


# DIGITAL LOGIC AND DESIGN

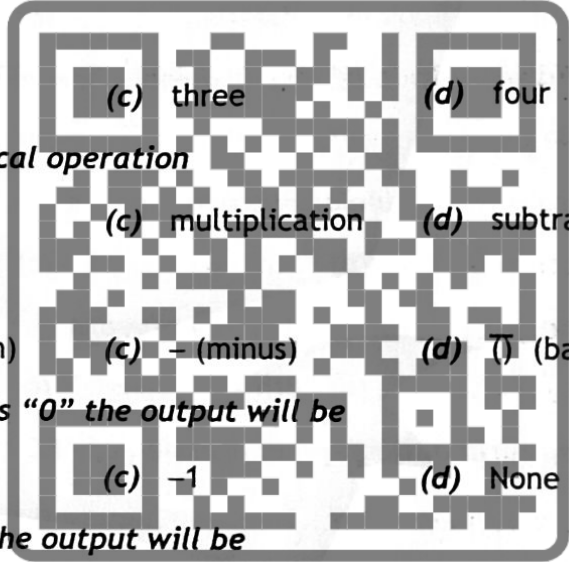


# Multiple Choice Questions (M.C.Qs)

Choose the right answer:

1. The universal gate is  
(a) NAND Gate (b) AND Gate (c) OR Gate (d) None of these
2. The \_\_\_\_\_ is inverter  
(a) AND (b) OR Gate (c) Not (d) None of these
3. In Boolean Algebra the bar (-) sign indicates \_\_\_\_\_  
(a) OR Operation (b) NOR Operation (c) NOT Operation (d) Both b and c
4. The Boolean algebra is used for \_\_\_\_\_  
(a) Creating circuits (b) apply 12 rules of Boolean  
(c) Simplify the Boolean expression (d) Differentiate the gates
5. With the combination of three variable how many outputs are expected altogether?  
(a) Three (b) Six (c) Eight (d) Nine
6.  $A+A=A$  is a \_\_\_\_\_ rule of Boolean algebra  
(a) 3rd (b) 5th (c) 6th (d) 7th
7.  $A.\bar{A}=0$  is a \_\_\_\_\_ rule of Boolean algebra  
(a) 1st (b) 8th (c) 6th (d) 10th
8. Simplified form of Boolean expression of  $ABC + ABC + \bar{A}B$  is \_\_\_\_\_  
(a) A (b)  $\bar{B}$  (c) C (d)  $\bar{B}$
9. Simplified form of Boolean expression of  $ABC + \bar{A}BC$  is \_\_\_\_\_  
(a) AC (b) BC (c) B (d)  $\bar{A}$
10. Boolean expression of the given circuit is  
  
 (a)  $(\bar{A}+\bar{B})$  (b) B.A (c) A.B (d)  $\bar{A.B}$
11. A digital signal in electronic circuit is represented in \_\_\_\_\_ form  
(a) binary (b) octal (c) decimal (d) hexa
12. Binary "0" is also called as  
(a) True (b) False (c) No Value (d) Void
13. Binary "1" is also called as  
(a) True (b) False (c) No Value (d) Void

- 14.** Electronic circuits in a digital system are called as  
 (a) logic gates (b) Entry Point (c) Stationary Point (d) None of These
- 15.** Logic gates are divided into \_\_\_\_ categories  
 (a) One (b) Two (c) Three (d) Four
- 16.** AND Gate is a \_\_\_\_ gate  
 (a) Universal (b) standard (c) basic (d) secondary
- 17.** OR Gate is a \_\_\_\_ gate  
 (a) Universal (b) standard (c) basic (d) secondary
- 18.** NOT Gate is a \_\_\_\_ gate  
 (a) Universal (b) standard (c) basic (d) secondary
- 19.** Tabular representation of input and output values within a logic circuit is termed as  
 (a) Logic Gate (b) Output table (c) Input table (d) Truth table
- 20.** Minimum inputs for AND gate is  
 (a) one (b) two (c) three (d) four
- 21.** AND gate operates \_\_\_\_ mathematical operation  
 (a) addition (b) Inverter (c) multiplication (d) subtraction
- 22.** AND operation is denoted by  
 (a)  $\cdot$  (dot) (b)  $+$  (addition) (c)  $-$  (minus) (d)  $\bar{()}$  (bar)
- 23.** In AND gate if any of the two input is "0" the output will be  
 (a) 1 (b) 0 (c) -1 (d) None of these
- 24.** In AND gate if both inputs are "1" the output will be  
 (a) 1 (b) 0 (c) -1 (d) None of these
- 25.** Minimum inputs for OR gate is  
 (a) one (b) two (c) three (d) four
- 26.** OR gate operates \_\_\_\_ mathematical operation  
 (a) addition (b) Inverter (c) multiplication (d) subtraction
- 27.** OR operation is denoted by  
 (a)  $\cdot$  (dot) (b)  $+$  (addition) (c)  $-$  (minus) (d)  $\bar{()}$  (bar)
- 28.** In OR gate if any of the two input is "0" the output will be  
 (a) 1 (b) 0 (c) -1 (d) None of these
- 29.** In OR gate if both inputs are "1" the output will be  
 (a) 1 (b) 0 (c) -1 (d) None of these



30. Minimum inputs for NOT gate is

- (a) one (b) two (c) three (d) four

31. NOT gate operates \_\_\_\_\_ mathematical operation

- (a) addition (b) Inverter (c) multiplication (d) subtraction

32. NOT operation is denoted by

- (a) . (dot) (b) + (addition) (c) - (minus) (d)  $\overline{\quad}$  (bar)

33. In NOT gate if the input is "0" the output will be

- (a) 1 (b) 0 (c) -1 (d) None of these

33. In NOT gate if the input is "0" the output will be

- (a) 1 (b) 0 (c) -1 (d) None of these

34. In NOT gate if the input is "1" the output will be

- (a) 1 (b) 0 (c) -1 (d) None of these

35. A gate which can implement any Boolean function without the need to use basic gate is called as

- (a) Basic (b) Primary (c) Universal (d) Standard

36. NAND gate is a

- (a) Basic gate (b) Primary gate (c) Universal gate (d) Standard gate

37. NOR gate is a

- (a) Basic gate (b) Primary gate (c) Universal gate (d) Standard gate

38. NAND gate is a combination of two basic gates which are

- (a) NOT-AND (b) OR-AND (c) NOT-OR (d) AND-OR

39. NOR gate is a combination of two basic gates which are

- (a) NOT-AND (b) OR-AND (c) NOT-OR (d) AND-OR

40.  $Y = (\overline{A}B)$  is a Boolean expression for

- (a) NOR (b) AND (c) NOT (d) NAND

41. Minimum inputs for NAND gate is

- (a) one (b) two (c) three (d) four

42. In NAND gate if any of the two input is "1" the output will be

- (a) 1 (b) 0 (c) -1 (d) None of these

43. In NAND gate if both of the two input are "0" the output will be

- (a) 1 (b) 0 (c) -1 (d) None of these

JOIN  
FOR  
MORE!!!

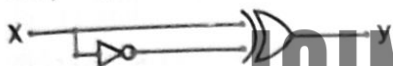


44.  $Y = (\overline{A+B})$  is a Boolean expression for  
 (a) NOR (b) AND (c) NOT (d) NAND
45. Minimum inputs for NOR gate is  
 (a) one (b) two (c) three (d) four
46. In NOR gate if any of the two input is "0" the output will be  
 (a) 1 (b) 0 (c) -1 (d) None of these
47. In NOR gate if both of the two inputs are "1" the output will be  
 (a) 1 (b) 0 (c) -1 (d) None of these
48. Boolean algebra was invented in  
 (a) 1854 (b) 1954 (c) 1956 (d) 1856
49. Boolean algebra was invented by  
 (a) George Boole (b) George Boolean (c) Kim Boolean (d) Al-khawarizm
50. There are \_\_\_\_\_ rules to simplify Boolean gates  
 (a) 12 (b) 21 (c) 11 (d) 14
51. Pre-defined rules to simplify Boolean expressions are called as  
 (a) Laws of algebra (b) Laws of Boolean algebra expressions  
 (c) Laws of gates (d) Laws of Arithmetic
52. According to laws of Boolean algebra  $A + 0 =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1
53. According to laws of Boolean algebra  $A + 1 =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1
54. According to laws of Boolean algebra  $A \cdot 0 =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1
55. According to laws of Boolean algebra  $A \cdot 1 =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1
56. According to laws of Boolean algebra  $A + A =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1
57. According to laws of Boolean algebra  $A + \overline{A} =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1
58. According to laws of Boolean algebra  $A \cdot \overline{A} =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\overline{A}$  (d) 1



59. According to laws of Boolean algebra  $A \cdot \bar{A} =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\bar{A}$  (d) 1
60. According to laws of Boolean algebra  $\bar{\bar{A}} =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\bar{A}$  (d) 1
61. According to laws of Boolean algebra  $A + A \cdot B =$  \_\_\_\_\_  
 (a) A (b) 0 (c)  $\bar{A}$  (d) 1
62. According to laws of Boolean algebra  $A + \bar{A} \cdot B =$  \_\_\_\_\_  
 (a)  $A+B$  (b)  $A \cdot B$  (c)  $\bar{A}$  (d) B
63. According to laws of Boolean algebra  $(A+B)(A+C) =$  \_\_\_\_\_  
 (a)  $A+AC$  (b)  $B+AC$  (c)  $\bar{A}+BC$  (d)  $A+BC$

64. Output for following gate is



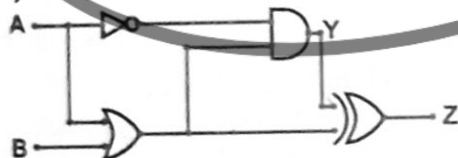
- (a) X (b) 0 (c)  $\bar{X}$  (d) 1

65. Output for following gate is represented by



- (a) OR (b) NOR (c) AND (d) NAND

66. If  $A=1$ ,  $B=1$  what will be the values of Y and Z



- (a)  $Y=0, Z=0$  (b)  $Y=1, Z=0$  (c)  $Y=0, Z=1$  (d)  $Y=1, Z=1$

## Answers

1. (a)	2. (c)	3. (d)	4. (c)	5. (c)	6. (b)	7. (b)	8. (b)
9. (b)	10. (d)	11. (a)	12. (b)	13. (a)	14. (a)	15. (b)	16. (c)
17. (c)	18. (c)	19. (d)	20. (b)	21. (c)	22. (a)	23. (b)	24. (a)
25. (a)	26. (c)	27. (b)	28. (b)	29. (a)	30. (a)	31. (b)	32. (d)
33. (a)	34. (b)	35. (c)	36. (c)	37. (c)	38. (a)	39. (c)	40. (d)
41. (b)	42. (b)	43. (a)	44. (a)	45. (b)	46. (a)	47. (b)	48. (b)
49. (a)	50. (a)	51. (b)	52. (a)	53. (d)	54. (b)	55. (a)	56. (a)
57. (d)	58. (a)	59. (b)	60. (a)	61. (c)	62. (a)	63. (d)	64. (d)
65. (b)	66. (c)						

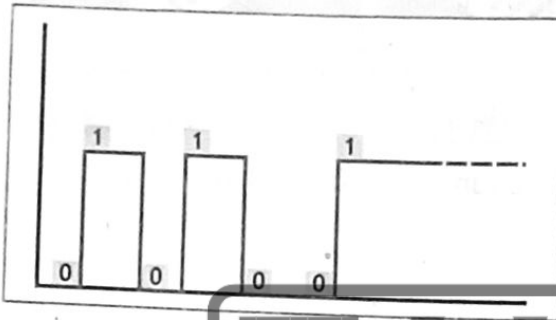
## Short & Detailed Answer Questions

**Q.1** What is a logic gate?

**Ans.** Logic gate are the electronic circuits in a digital system. Logic gate performs logic operations such as AND, OR, etc.

**Q.2** What is a binary logic gate?

**Ans.** A binary logic gate consists of binary variables and logical operations. The variables are denoted by letters.



**Q.3** How many values a binary logic gate can have?

**Ans.** A binary logic gate can have two possible values i.e 0 and 1.

**Q.4** What are the different names for 0 and 1?

**Ans.** 0 and 1 are also called as True (1) and False (0).

**Q.5** What are the different categories for logic gates?

**Ans.** Logic gates are divided into two categories:

- Basic Logic gates
- Universal Logic gates

**Q.6** What is a basic logic gate?

**Ans.** A basic logic gate is one which is used to perform basic logical operations like AND, OR, and NOT.

**Q.7** What is a truth table?

**Ans.** A truth table is a tabular representation of all the combinations of values for inputs and their corresponding outputs.

**Q.8** Name the basic logic gates?

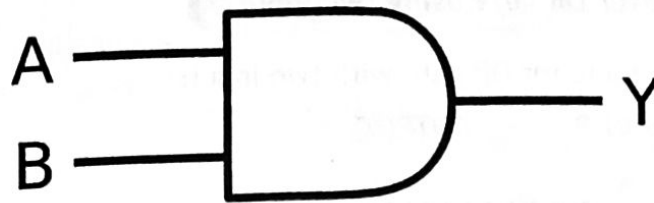
**Ans.** There are three basic logic gates:

- AND Gate
- OR Gate
- NOT Gate



**Q.9** What is AND Gate? OR What logical operation is performed with AND Gate?

**Ans.** An AND gate is a digital circuit that has two or more inputs and one output. It operates on logical multiplication rules.



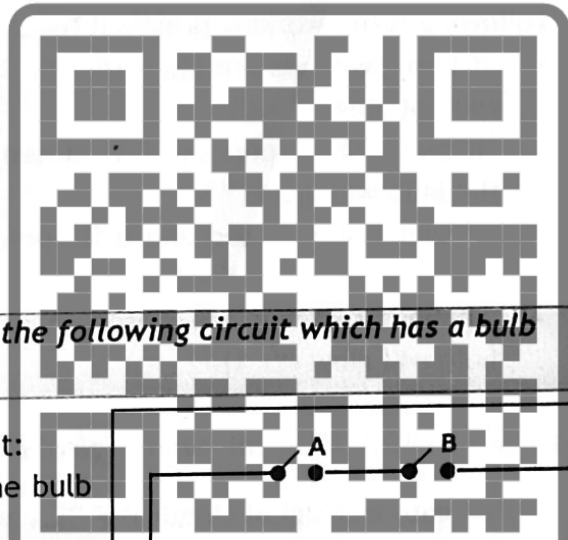
**Q.10** What is the Boolean expression for AND gate?

**Ans.** Boolean expression for AND Gate is  $Y = A \cdot B$ .

**Q.11** Create a truth table for AND gate using two inputs?

**Ans.** Following is the truth table for AND gate with two inputs:

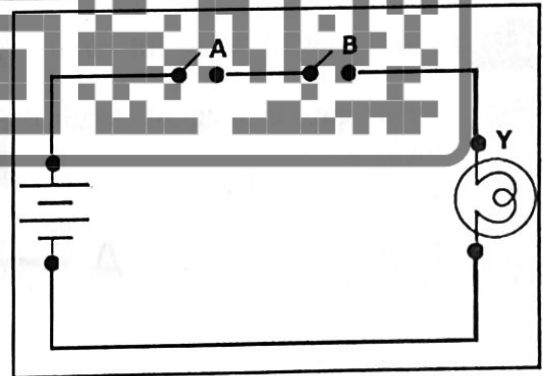
INPUT A	INPUT B	OUTPUT
A	A	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1



**Q.12** Explain the working principal of AND gate in the following circuit which has a bulb attached to the output?

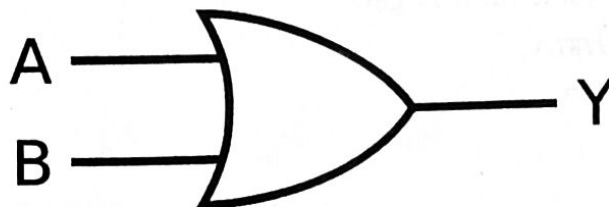
**Ans.** Following is the working principal for this circuit:

- If both switches are open ( $A = 0, B = 0$ ) the bulb will not glow ( $Y = 0$ ).
- If switch A is open and B is closed ( $A = 0, B = 1$ ) the bulb will not glow ( $Y = 0$ ).
- If switch A is closed and B is open ( $A = 1, B = 0$ ) the bulb will not glow ( $Y = 0$ ).
- If both switches are closed ( $A = 1, B = 1$ ) the bulb will glow ( $Y = 1$ ).



**Q.13** What is OR Gate? OR What logical operation is performed with OR Gate?

**Ans.** An OR gate is a digital circuit that has two or more inputs and one output. It operates on logical addition rules.



**Q. 14** What is the Boolean expression for OR gate?

**Ans.** Boolean expression for OR Gate is  $Y=A+B$ .

**Q. 15** Create a truth table for OR gate using two inputs?

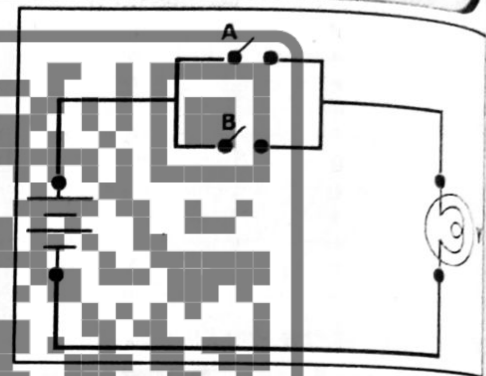
**Ans.** Following is the truth table for OR gate with two inputs:

INPUT A	INPUT B	OUTPUT
A	A	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

**Q. 16** Explain the working principal of OR gate in the following circuit which has a bulb attached to the output?

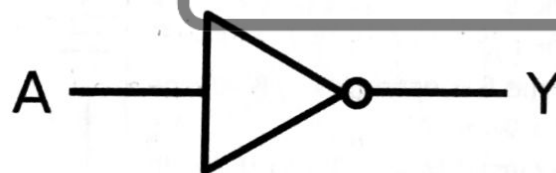
**Ans.** Following is the working principal for this circuit:

- If both switches are open ( $A = 0, B = 0$ ) the bulb will not glow ( $Y = 0$ ).
- If switch A is open and B is closed ( $A = 0, B = 1$ ) the bulb will glow ( $Y = 1$ ).
- If switch A is closed and B is open ( $A = 1, B = 0$ ) the bulb will glow ( $Y = 1$ ).
- If both switches are closed ( $A = 1, B = 1$ ) the bulb will glow ( $Y = 1$ ).



**Q. 17** What is NOT Gate? OR What logical operation is performed with NOT Gate?

**Ans.** A NOT gate is a digital circuit that has a single input and a single output. It is also called as inverter.



**Q. 18** What is the Boolean expression for NOT gate?

**Ans.** Boolean expression for NOT gate is  $Y=\bar{A}$ .

**Q. 19** Create a truth table for NOT gate?

**Ans.** Following is the truth table for NOT gate:

INPUT A	OUTPUT
A	$Y=\bar{A}$
0	1
1	0



**Q 20** Explain all the three basic logic gates with their graphical representation?

**Ans.** There are three basic gates:

**AND Gate:**

It has at least two inputs and one output. It works on the logical multiplication function.



**OR Gate:**

It has at least two inputs and one output. It works on the logical addition function.



**NOT Gate:**

It has one input and one output. It is also called as inverter.



**Q 21** Draw a truth table for AND gate with 3 inputs?

**Ans.** Following is the truth table for AND gate with 3 inputs

A	B	C	$Y=A.B.C$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

**Q 22** Draw a truth table for OR gate with 3 inputs?

**Ans.** Following is the truth table for OR gate with 3 inputs:

A	B	C	$Y=A+B+C$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

**Q 23** Differentiate between AND gate and OR gate?

**Ans.** Following table shows differentiation between AND gate and OR gate:



AND Gate	OR Gate
It executes logical multiplication.	It executes logical addition.
Boolean expression is $Y = A \cdot B$	Boolean expression is $Y = A + B$
Logic high output is achieved when necessarily both the inputs are high	Logic high is achievable in case of even single high input.

**Q.24** What is universal logic gate?

**Ans.** A universal logic gate is one which can implement any Boolean function without the need to use any other type of basic gates.

**Q.25** What are some of the universal gates?

**Ans.** Following are the universal gates:

- NAND
- NOR

**Q.26** What is NAND gate?

**Ans.** A NAND gate is a combination of two gates i.e Not-AND. Both gates are connected in series.

**Q.27** What is the Boolean expression for NAND gate?

**Ans.** Boolean expression for NAND gate is  $Y = \overline{A \cdot B}$ .

**Q.28** Write down truth table for NAND gate using two inputs?

**Ans.** Following is the truth table for NAND gate with two inputs:

INPUT A	INPUT B	OUTPUT
A	A	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

**Q.29** What is NOR gate?

**Ans.** A NOR gate is a combination of two basic gates i.e Not-OR. The output of NOR gate is inverse of OR.

**Q.30** Write down truth table for NOR gate using two inputs?

**Ans.** Following is the truth table for NOR gate with two inputs:

INPUT A	INPUT B	OUTPUT
A	A	$Y = \overline{A + B}$

0	0	0
0	1	1
1	0	1
1	1	1

**Q.31** What is the Boolean expression for NOR gate?

**Ans.** The Boolean expression for NOR gate is  $Y = \overline{A+B}$ .

**Q.32** Differentiate between NAND and NOR gate?

**Ans.** Following table shows differentiation between AND gate and OR gate:

NAND Gate	NOR Gate
It is a combination of Not and AND gate	It is a combination of Not and OR gate
Boolean expression is $Y = \overline{A \cdot B}$	Boolean expression is $Y = \overline{A+B}$
Logic high output is achieved when necessarily both the inputs are NOT high	Logic high output is achieved when necessarily both the inputs are low

**Q.33** Differentiate between basic and universal logic gates?

**Ans.** Following is the difference between universal and basic gate:

BASIC GATE	UNIVERSAL GATE
They are not derived gates	They are formed with the combination of two basic gates
They perform basic logical operations such as multiplication, addition	They perform invert operations
Examples are AND, OR, NOT	Examples are NAND, NOR

**Q.34** What is Boolean Algebra?

**Ans.** Boolean algebra is a branch of mathematics that deals with operations on logical values with binary variables.

**Q.35** Who invented Boolean algebra?

**Ans.** Boolean algebra was invented by George Boole.

**Q.36** When was Boolean algebra invented?

**Ans.** Boolean algebra was invented in 1847.

**Q.37** What are rules for Boolean Algebra?

**Ans.** Boolean algebra rules are pre-defined rules that help to simplify logical expression.

**Q.38** How many basic Boolean algebra rules exist?

**Ans.** There are 12 basic rules for Boolean algebra which helps in solving logical expressions.

**Q.39** What are the 12 Boolean algebra rules? OR Write down list of Boolean algebra rules?

**Ans.** Following are the 12 Laws of Boolean algebra:

Rules No	Rule
1	$A + 0 = A$
2	$A + 1 = 1$
3	$A \cdot 0 = 0$
4	$A \cdot 1 = A$
5	$A + A = A$
6	$A + \bar{A} = 1$
7	$A \cdot A = A$
8	$A \cdot \bar{A} = 0$
9	$\bar{\bar{A}} = A$
10	$A + A \cdot B = A$
11	$A + \bar{A} \cdot B = A + B$
12	$(A + B)(A + C) = A + BC$

**Q.40** Explain the purpose of inverter?

**Ans.** Inverter is used to change the output of a logic gate. If the output is "0" it will be converted to "1" and if the output is "1" it will be converted to "0".

**Q.41** Simplify the following Boolean expressions?

- A.  $AB + A\bar{B} = A$
- B.  $(A+B) + (A+\bar{B}) = A$
- C.  $B = BC + B\bar{C} + BA$
- D.  $AC + AB\bar{C} = A\bar{C}$
- E.  $(A+B)(A+C)$
- F.  $AB + A(B+C) + B(B+C)$

**Ans.**

**A.**  $AB + A\bar{B} = A$

Taking L.H.S

$$AB + A\bar{B}$$

Taking "A" as common

$$A(B + \bar{B})$$

$$A(1)$$

$$A \cdot 1 = A$$

Hence LHS=RHS

$$B + \bar{B} = 1$$



$$(A+B) + (A+\bar{B}) = A$$

Taking LHS

**B.**  $(A+B) + (A+\bar{B})$

$$AA + A\bar{B} + BA + B\bar{B}$$

$$A + A\bar{B} + BA + 0$$

$$A + A(\bar{B} + B)$$

$$A + A(1)$$

$$A = A$$

Hence LHS = RHS

Using  $A.A = A$  &  $A.\bar{A} = 0$

$$(B + \bar{B}) = 1$$

$$A + A = A$$

**C.**  $B = BC + B\bar{C} + BA$

Taking C Common on RHS

$$B(C + \bar{C}) + BA$$

$$B.1 + BA$$

$$B(1 + A)$$

$$B = B$$

Hence LHS = RHS

**D.**  $A\bar{C} + ABC = A\bar{C}$

Taking LHS

$$A\bar{C} + ABC$$

$$A\bar{C}(1 + B) \quad 1 + A = 1$$

$$A\bar{C}(1)$$

$$A\bar{C}$$

Hence LHS = RHS

**E.**  $(A+B)(A+C) = A + BC$

Taking LHS

$$(A+B)(A+C)$$

$$AA + AC + AB + BC$$

$$A + AC + AB + BC \quad A.A = A$$

Taking A as common

$$A(1 + C) + AB + BC$$

$$A(1) + AB + BC \quad 1 + A = A$$

Taking A common again

$$A(1 + B) + BC$$

$$A(1) + BC \quad 1 + A = A$$

$$A + BC = A + BC$$

Hence LHS = RHS

**F.**  $AB + A(B+C) + B(B+C)E$

$$AB + AB + AC + B(B+C)$$

$$AB + AC + BC + BB \quad BB = B$$

$$AB + AC + BC + B$$

$$AC + AB + B(1 + C) \quad 1 + C = 1$$

$$AC + B(A + 1)$$

$$AC + B$$

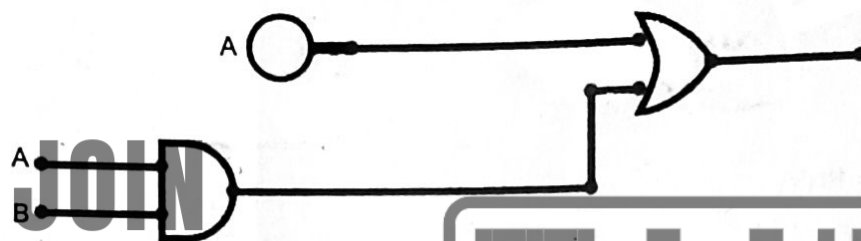


**Q.42** Draw logic circuit of the following Boolean expressions?

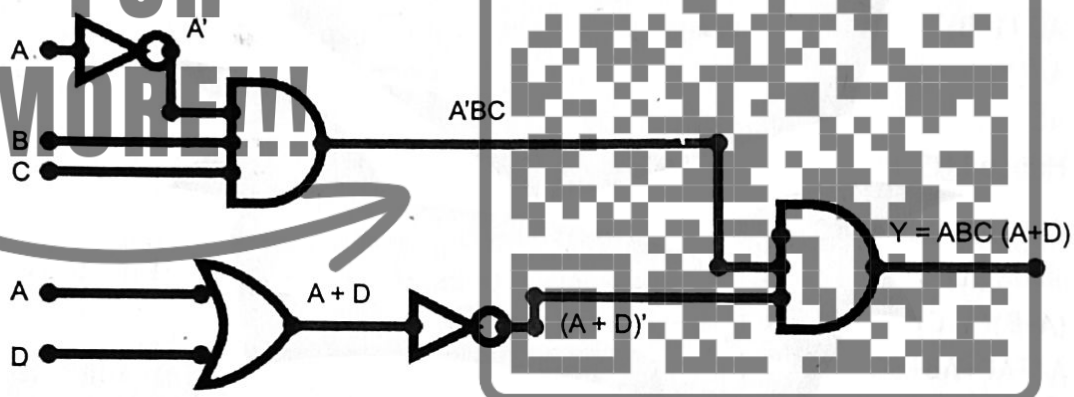
- A.  $A = A + A.B$
- B.  $Y = ABC(\overline{A+D})$
- C.  $X = \overline{AB}(\overline{C+D})$
- D.  $Q = (A.B) + (\overline{A+B})$
- E.  $Q = (A.B) + (A.B) + (\overline{A+B})$
- F.  $Q = (A+B)' + (A.B)'$

**Ans.**

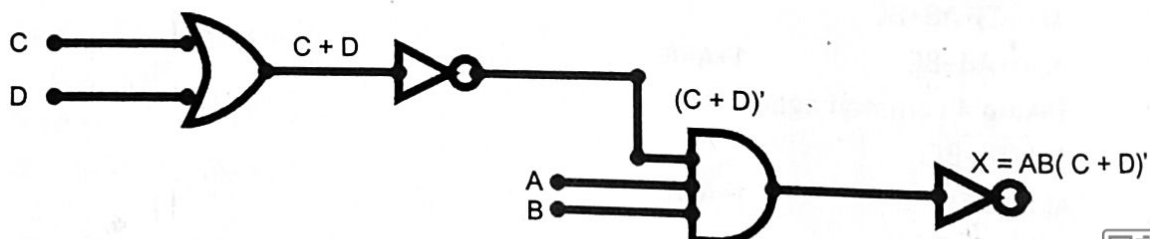
**A.**  $A = A + A.B$



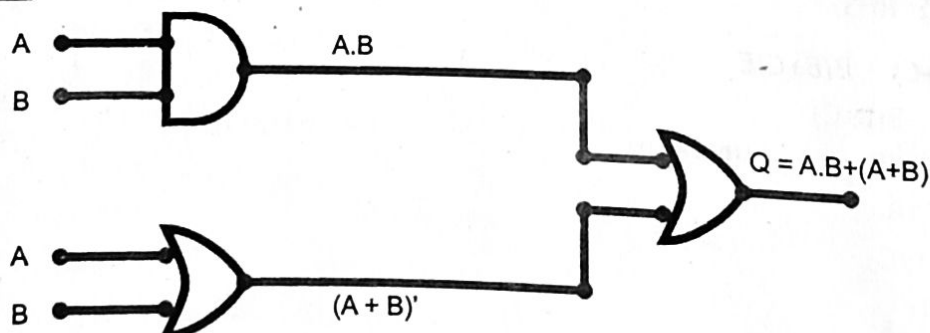
**B.**  $Y = ABC(\overline{A+D})$



**C.**  $X = \overline{AB}(\overline{C+D})$

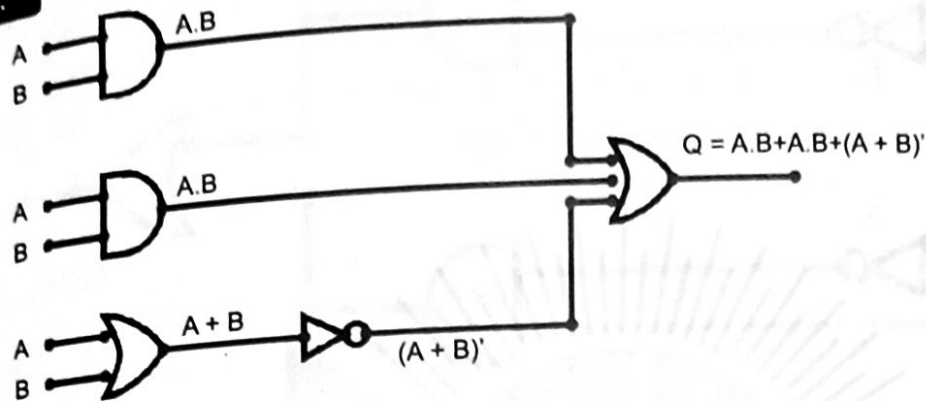


**D.**  $Q = A.B + (\overline{A+B})$

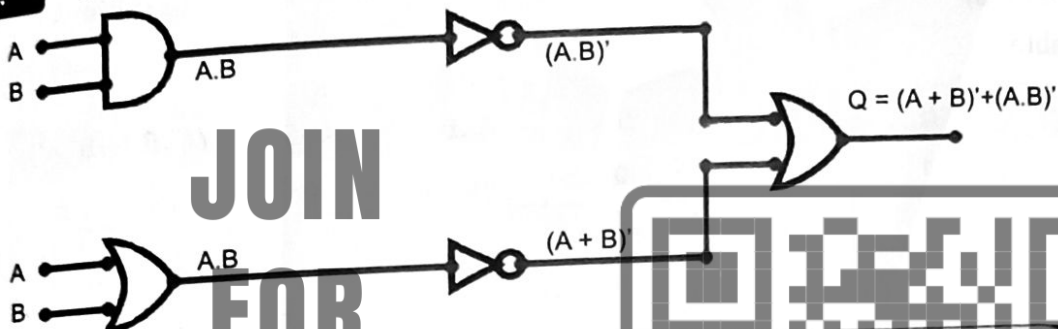




E.  $Q = (A.B) + (A.B) + (\overline{A+B})$

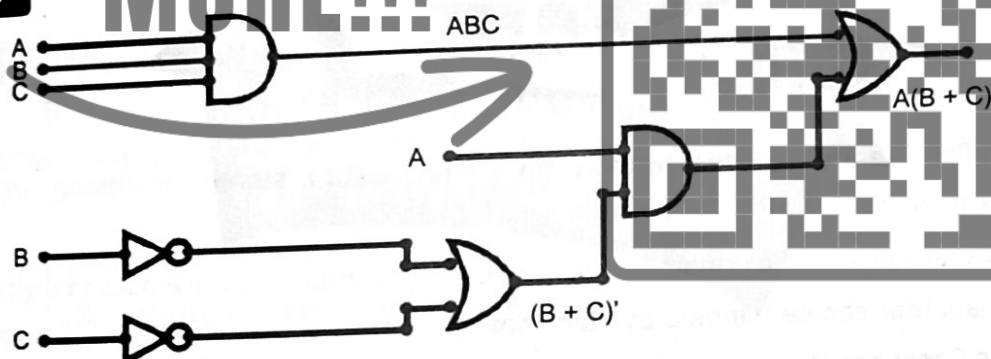


F.  $Q = (A+B)' + (A.B)'$



Q.43 Derive Boolean expressions from the given circuits and make truth table:

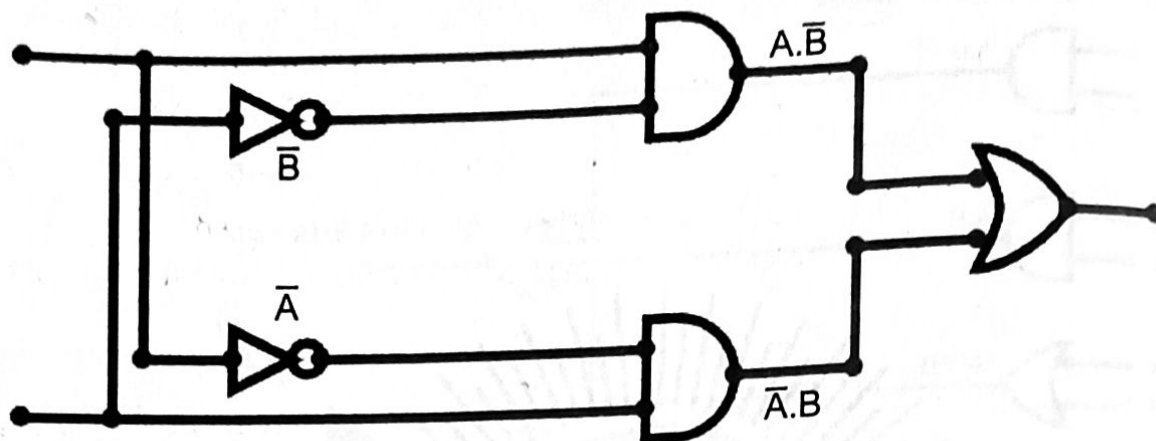
A.



Ans. A.  $Q = ABC + A(B' + C')$

Truth Table

A	B	C	ABC	B'	C'	A(B' + C')	ABC + A(B' + C')
1	1	1	1	0	0	0	1
0	1	1	0	0	0	0	0
1	0	1	0	1	0	1	1
0	0	1	0	1	0	0	0
1	1	0	0	0	1	1	1
0	1	0	0	0	1	0	0
1	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0



Ans.  $Q = (A.B') + (A'.B)$

Truth Table

A	B	A'	B'	A'.B'	A'.B'	(A'.B')+(B'.A')
0	1	1	0			
0		1	1			
1	1	0	0			
1	0	0	1			

JOIN  
FOR  
MORE!!!

## Summary

- Data Representation refers to the form in which data is stored, processed, and transmitted.
- Digital devices such as smart phones, iPods, and computers store data in digital formats that can be handled by electronic circuitry.
- Logic Gates are the electronic circuits in a digital system. Logical Gates perform logical operations like AND, OR, NOT, NAND, NOR etc.
- The logic gate is the basic unit of digital logic circuits, there are mainly three basic gates AND, OR, and NOT and these logical gates perform AND, OR, and NOT operation in the digital system.
- An AND gate is a digital circuit that has two or more inputs and a single output. AND gate operates on logical multiplication rules.
- Boolean Expression of AND gate:  $Y = A.B$  An OR gate is a digital circuit that has two or more inputs and a single output. OR gate operates on logical Addition rules.
- Boolean expression of OR gate is  $Y = A + B$ .
- A NOT gate is a digital circuit that has a single input and a single output. It is also known as INVERTER.

Universal Gates are logic gates. They are capable of implementing any Boolean function without requiring any other type of gate. There are two types of universal gates

A NAND Gate could be construct by connecting a NOT Gate at the Output terminal of the AND Gate. Boolean expression of NAND gate is  $Y = (A.B)'$  or  $Y = AB$ .

A NOR Gate could construct by connecting a NOT Gate at the output terminal of OR Gate.

The Boolean expression of NOR gate is  $Y = (A+B)'$  or  $Y = A + B$ .

The Boolean arithmetic rules are pre-defined rules that help to simplify the logical expression. There are 12 Boolean algebra rules.

A NOR Gate could construct by connecting a NOT Gate at the output terminal of OR Gate.

The Boolean expression of NOR gate is  $Y = (A+B)'$  or  $Y = A + B$ .

The Boolean arithmetic rules are pre-defined rules that help to simplify the logical expression. There are 12 Boolean algebra rules.

## Solution of Textbook Exercise

### A. Encircle the correct answer:

Ans. See "Multiple Choice Question (MCQS)" Q1 – 10

### B. Respond the following:

1. Explain all basic logic gates and their operations?

Ans. See "Short/Detailed Questions and Answers" – Q.20

2. Differentiate between NAND and NOR gates?

Ans. See "Short/Detailed Questions and Answers" – Q.32

3. Why do we use Boolean Algebra?

Ans. See "Short/Detailed Questions and Answers" – Q.32

4. Explain the function of Inverter?

Ans. See "Short/Detailed Questions and Answers" – Q.41

5. Explain the purpose of Truth table?

Ans. See "Short/Detailed Questions and Answers" – Q.7

6. Simplify the following Boolean expression?

$$Z = AB + A(B+C) + B(B+C)$$

Ans. See "Short/Detailed Questions and Answers" – Q.42(f)



## Lab Activity

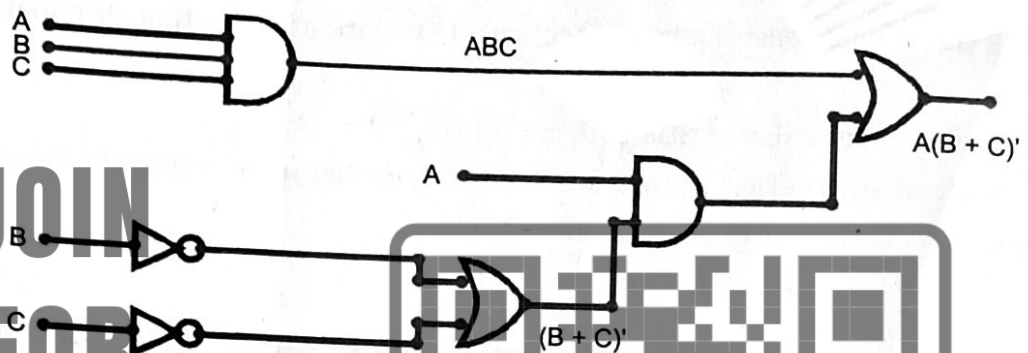
1. Draw a Logic Circuit of 10th Law of Boolean Algebra.

Ans. See "Short/Detailed Questions and Answers"– Q.43(a)

2. Design a Logic Circuit from Boolean expression  $Q = (A.B) + (A.B) + (\overline{A+B})$ .

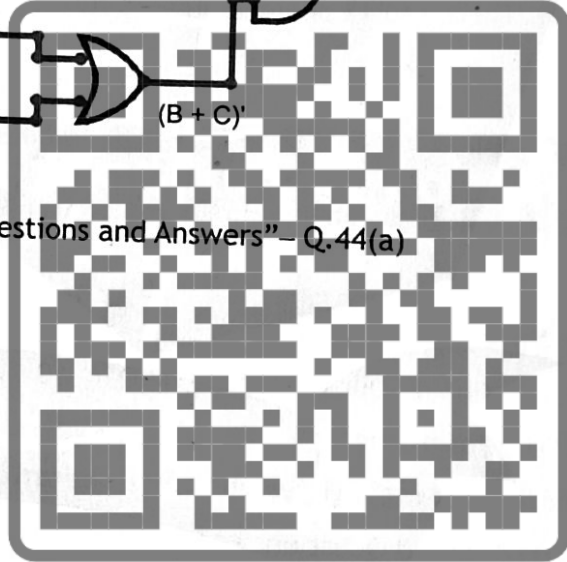
Ans. See "Short/Detailed Questions and Answers"– Q.43(e)

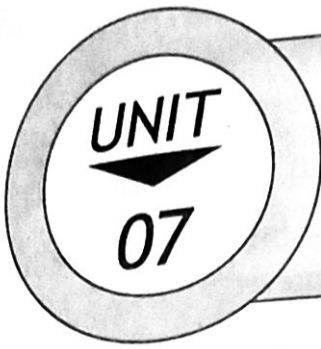
3. Derive the Boolean expression from the given circuit and make truth table and simplify Boolean expression



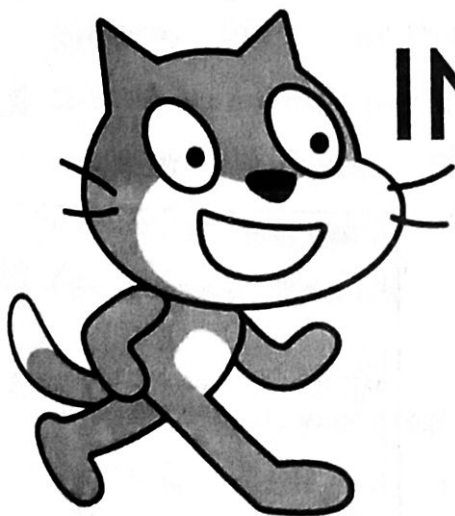
Ans. See "Short/Detailed Questions and Answers"– Q.44(a)

JOIN FOR MORE!!!





## INTRODUCTION TO SCRATCH



# INTRODUCTION TO SCRATCH

A Simple Programming  
Language

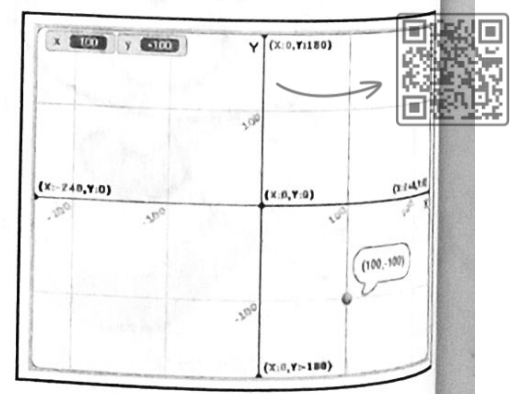
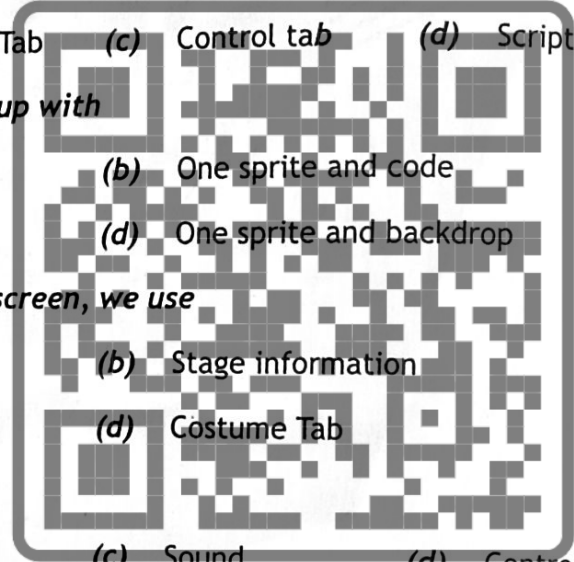




# Multiple Choice Questions (M.C.Qs)

Choose the right answer:

1. The feature of scratch is
  - (a) It is a visual
  - (b) Its free forever
  - (c) No need to remember coded
  - (d) All of the above
2. In scratch the character which moves on the stage is called a
  - (a) Sprite
  - (b) Command
  - (c) Script
  - (d) event
3. Repeat 10, forever and if ... then codes are available in
  - (a) Motion
  - (b) Control
  - (c) Look
  - (d) Sensing
4. The looks of sprite can be changed by using
  - (a) Backdrop tab
  - (b) Costume Tab
  - (c) Control tab
  - (d) Script Tab
5. A scratch program is at least made up with
  - (a) Many sprites
  - (b) One sprite and code
  - (c) One sprite
  - (d) One sprite and backdrop
6. To change the position of sprite on screen, we use
  - (a) Coordinates
  - (b) Stage information
  - (c) Command pallets
  - (d) Costume Tab
7. Move 10 steps can be found under
  - (a) Look
  - (b) Motion
  - (c) Sound
  - (d) Control
8. Turn command turn sprite to specified
  - (a) Coordinates
  - (b) Steps
  - (c) Degree
  - (d) Seconds
9. The repeat 10 command is available in
  - (a) Events
  - (b) Control
  - (c) Looks
  - (d) Motion
10. In the given picture identify the x and y coordinates
  - (a)  $x=100$ ,  $y=-100$
  - (b)  $x=-100$ ,  $y=100$
  - (c)  $x=100$ ,  $y=100$
  - (d)  $x=-100$ ,  $y=-100$
11. FORTRAN was emerged in
  - (a) 1947
  - (b) 1945
  - (c) 1950
  - (d) 1961



Q in "2D" stands for

- (a) Dimensional (b) Direction (c) Distance (d) Delay

Scratch was developed by

- (a) Microsoft (b) Tesla (c) IBM (d) MIT

Scratch has \_\_\_\_\_ fee associated with it for usage

- (a) \$100 (b) \$25 (c) \$0 (d) \$50

Scratch can be used

- (a) Online and Offline (b) Only Online  
(c) Only Offline (d) On Cloud

Sprites are the images of

- (a) Object (b) Cartoons (c) Characters (d) All of these

\_\_\_\_\_ is the least number of sprites required in Scratch

- (a) 2 (b) 1 (c) 0 (d) 3

\_\_\_\_\_ is the default sprite in scratch

- (a) Cat (b) Dog (c) Bat (d) Bird

\_\_\_\_\_ is used to make sprites perform the task

- (a) sprint (b) script (c) code (d) editor

Output of your code can be visible on

- (a) stage (b) script (c) sprint (d) none of these

The main working area where sprite moves and performs action is called

- (a) stage (b) script (c) sprint (d) none of these

Stage is divided into

- (a) X and Y axis (b) Rows (c) Columns (d) Z axis

Co-ordinates are displayed at the

- (a) Bottom right corner (b) Bottom left corner  
(c) Top right corner (d) Top left corner

Co-ordinates indicate the

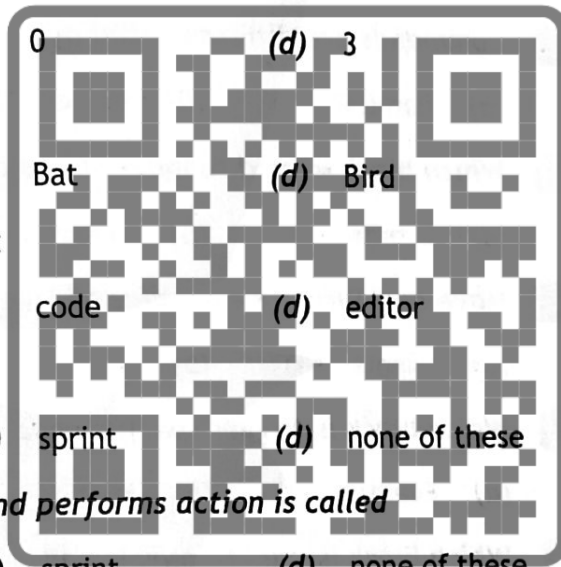
- (a) cursor (b) script (c) stage (d) sprite

Area which shows your program in scratch is called as

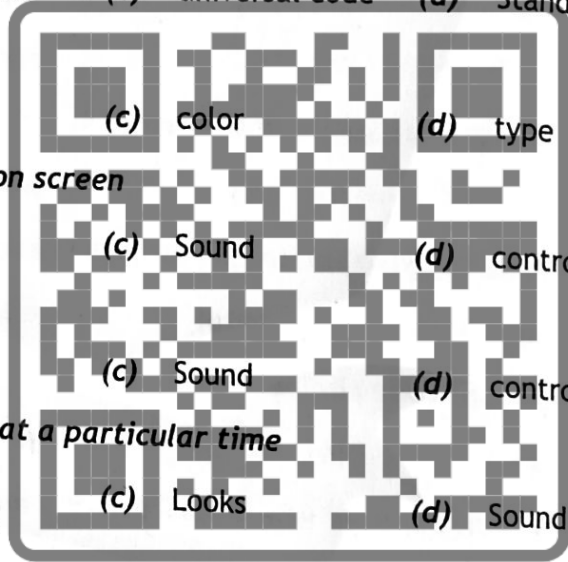
- (a) stage preview (b) stage area (c) sprite area (d) script area




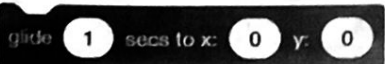
Sprite list displays

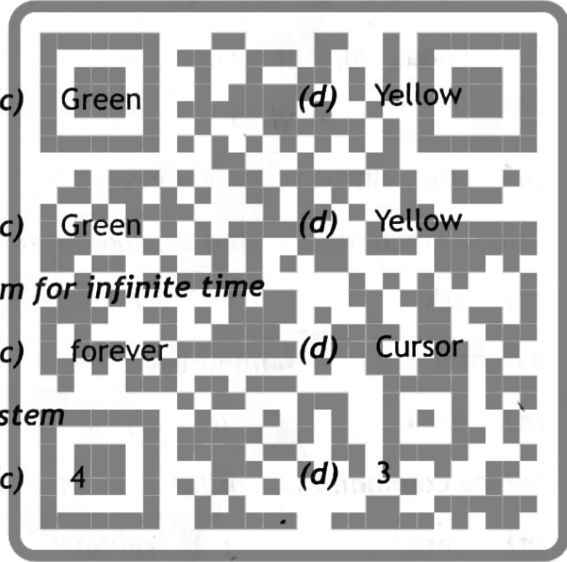
- (a) Thumbnails (b) Pictures (c) Code (d) location



27. Blue information icon can be clicked to change the \_\_\_\_\_ and behavior of sprite  
 (a) location (b) name (c) icon (d) color
28. Background of the stage is called as  
 (a) background (b) backdrop (c) projector (d) block
29. By default a scratch program has \_\_\_\_\_ backdrops  
 (a) 1 Backdrop (b) No Backdrop (c) multiple (d) Black
30. Script block has \_\_\_\_\_ tabs  
 (a) one (b) two (c) three (d) four
31. Script tab in scratch 3 is known as  
 (a) script block (b) script table (c) code tab (d) sprite tab
32. In scratch commands are shown as  
 (a) road blocks (b) code blocks (c) universal code (d) Standard code
33. Commands are differentiated by  
 (a) name (b) genre (c) color (d) type
34. Which block is used to move sprite on screen  
 (a) Events (b) Motion (c) Sound (d) control
35. Move and Turn are examples of  
 (a) Events (b) Motion (c) Sound (d) control
36. Which block is trigger specific code at a particular time  
 (a) Events (b) Sensing (c) Looks (d) Sound
37. Which block is used to play sound  
 (a) Events (b) Sensing (c) Looks (d) Sound
38. Which block is used to change the appearance of sprite and backdrop  
 (a) control (b) Sensing (c) Looks (d) Sound
39. Which block control the actions on the stage  
 (a) control (b) Sensing (c) Looks (d) Sound
40. Which block is used to sense any happening  
 (a) sensing (b) control (c) Looks (d) data
41. Which block is used to initialize variable  
 (a) sensing (b) control (c) Looks (d) data
42. Which block is used to draw lines, rectangles and other shape  
 (a) sensing (b) control (c) Looks (d) Pen



43. Which block is used to perform logical, arithmetic, and relational operators  
(a) sound (b) operator (c) Looks (d) data
44. Costume tab is used to change the  
(a) Costume of sprite (b) backdrop of sprite  
(c) icon of sprite (d) None of these
45. Sound tab is a  
(a) mandatory field (b) recommended field  
(c) optional field (d) fixed field
46. Cursor tools has \_\_\_\_ options  
(a) 1 (b) 2 (c) 4 (d) 5
47. Which flag is used to start the program  
(a) Red (b) Blue (c) Green (d) Yellow
48. Which flag is used to test the program  
(a) Red (b) Blue (c) Green (d) Yellow
49. Which flag is used to stop the program  
(a) Red (b) Blue (c) Green (d) Yellow
50. \_\_\_\_ command keeps repeating the program for infinite time  
(a) Test (b) Loop (c) forever (d) Cursor
51. Scratch has \_\_\_\_ dimensional coordinate system  
(a) 1 (b) 2 (c) 4 (d) 3
52.  is used to  
(a) triggers the upcoming code (b) triggers the previous code  
(c) triggers the current code (d) triggers a customized code
53.  is used to  
(a) turn 15 degrees (b) turns anticlockwise  
(c) turn specific degree (d) turns clockwise
54.  is used to  
(a) send sprite to x-axis (b) send sprite to y-axis  
(c) send sprite to specific location (d) send sprite to zero location
55.  is used to  
(a) glide sprite to x-axis (b) glide sprite to y-axis  
(c) glide sprite to specific location (d) glide sprite to zero location



56.  is used to

- (a) wait for specific time  
(b) wait only 1 sec  
(c) wait less than 1 sec  
(d) none of these

57.  is used to

- (a) repeat code block infinitely  
(b) repeat code block finitely  
(c) repeat code block once  
(d) repeat code block twice

58.  is used to

- (a) say specific phrase  
(b) speak specific phrase  
(c) play specific phrase  
(d) move cursor

59.  is used to

- (a) increase size of sprite  
(b) decrease size of sprite  
(c) change size of sprite  
(d) fix size of sprite

60.  is used to

- (a) draw line (b) draw circle  
(c) draw pen (d) draw triangle

61. Which command is used to take input in scratch

- (a) ask (b) say  
(c) answer (d) wait 1 sec

62. ask and sends command are available in which block

- (a) data (b) sound  
(c) sensing (d) control

63. set-to command is available in which block

- (a) data (b) sound  
(c) sensing (d) control

64. say command is available in which block

- (a) looks (b) sound  
(c) sensing (d) control

## Answers

1. (d)	2. (a)	3. (b)	4. (b)	5. (b)	6. (a)	7. (b)	8. (c)
9. (b)	10. (a)	11. (c)	12. (a)	13. (d)	14. (c)	15. (a)	16. (d)
17. (b)	18. (a)	19. (b)	20. (a)	21. (a)	22. (a)	23. (a)	24. (d)
25. (a)	26. (a)	27. (d)	28. (b)	29. (b)	30. (c)	31. (c)	32. (b)
33. (c)	34. (b)	35. (b)	36. (a)	37. (d)	38. (c)	39. (a)	40. (a)
41. (d)	42. (d)	43. (b)	44. (a)	45. (c)	46. (d)	47. (c)	48. (a)
49. (c)	50. (c)	51. (b)	52. (a)	53. (c)	54. (c)	55. (c)	56. (a)
57. (a)	58. (b)	59. (c)	60. (a)	61. (a)	62. (c)	63. (d)	64. (a)



# Short & Detailed Answer Questions

## Q.1 What is Scratch?

Ans. Scratch is a visual based programming language which does not require any code. Instead of code a program can be created by dragging and dropping some components.

## Q.2 Who developed Scratch?

Ans. Scratch was developed by Massachusetts Institute of Technology (MIT) Media lab.

## Q.3 What is the use of Scratch?

Ans. Scratch is used to create animations and games.

## Q.4 What are the important characteristics which a student learns by using scratch?

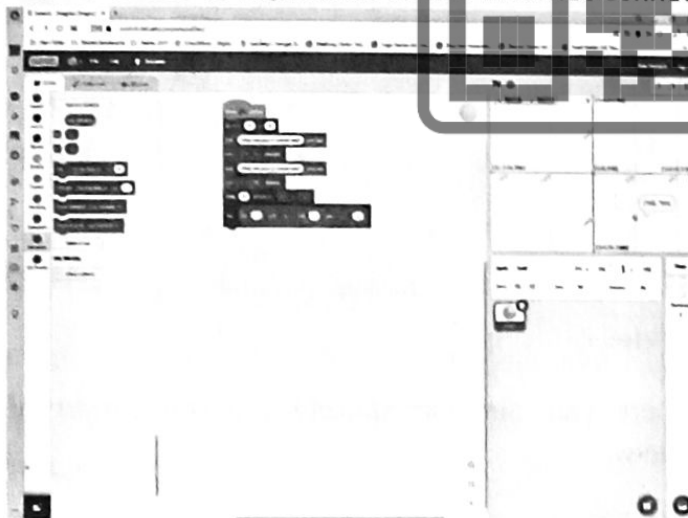
Ans. When a student uses scratch, they learn important mathematical and computing concepts, that improve their creative thinking, logical reasoning, problem solving, and collaboration.

## Q.5 How a student can use scratch?

Ans. Scratch is free to use and available in both online and offline mode.

## Q.6 What is the difference between online scratch and offline scratch?

Ans. Online scratch means that the language needs to have an active internet connection during processing and does not occupy and space on the local drive while offline scratch should be downloaded first in order to use it without an active internet connection.



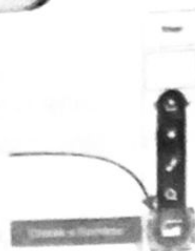
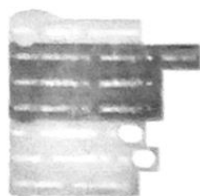
## Q.7 Write steps involved in downloading offline scratch?

Ans. Following are the steps involved in downloading offline scratch:

1. Download the offline editor
2. After downloading run the .exe file
3. First screen will appear which will ask drive location
4. It will also ask about the shortcuts that will be created

5. This will initiate the installation process

**Q.8 Define script, sprite, and backdrop?**



**Ans. Sprite:**

Sprites are the images of cartoons, characters, or objects. Cat is the default sprite.

**Scripts:**

Visual instructions which are used to create an animation, story or cartoon is called as script. Scripts are the instructions that perform specific task.

**Backdrop:**

It is the background that we assign to our stage.

**Q.9 What is a scratch editor?**

**Ans** Scratch editor is a space where all the components required to create an animation, or a story are available. It is further divided into blocks.

JOIN FOR MORE!!!



**Q.10 What is scratch preview?**

**Ans** It is the screen where you can immediately see the output of your codes. Project runs physically in this window.

**Q.11 How screen preview is divided?**

**Ans.** Screen preview is the windows where output is visible. It is divided into x (horizontal) and y (vertical) components.

**Q.12 What is script area?**

**Ans.** This area is used to show codes or programs. Blocks from the pallets can be dragged to this



area. Script area is also used to create scripts by placing blocks together.

**Q.13** What is sprite list?

**Ans.** It is a list of all the available sprites within a project. Each sprite is represented by a thumbnail.

**Q.14** What is Backdrop?

**Ans.** Backdrop is the background which is added to our stage. By default, there is no backdrop added in to the project.



**Q.15** What is script tab?

**Ans.** Script tab is also called as code tab in scratch 3 as it consists of different commands. This is the area which is considered as a toolbox.

**Q.16** Write down the purpose of following blocks.  
*Motion, Events, Sounds, Looks, Control, Sensing, Data, Pen, Operator?*

**Ans.** Following is the description for these blocks:

**Motion:**

It is used to move the sprite on the stage

**Events:**

It triggers specific code at a particular time

**Sounds:**

It is used to play sound

**Looks:**

It is used to change the appearance of sprite and backdrop



**Control:**

It control the actions on the stage

**Sensing:**

It sense any specific happening

**Data:**

It is used to initialize variables and list

**Pen:**

It is used to draw lines, and shapes

**Operators:**

It shows available arithmetic, logical and relational operator

**Q.17** Write the use of following codes (Forever, wait, say, play sound, go to x,y)?

**Ans.**

**Forever:**

It keeps repeating the following code infinite times

**Wait:**

It allows wait for specified seconds

**Say:**

It is used to says the given phrase

**Play sound:**

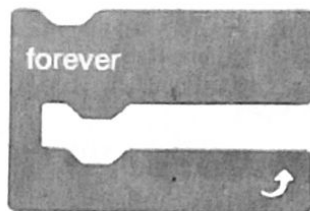
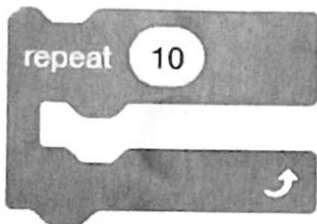
It is used to play the specified sound

**Go to x,y:**

It sends sprite to specified x and y coordinates



**Q.18** What is the difference between repeat 10 and forever?



**Ans.**

Repeat 10 and forever both are used to repeat the following code but repeat 10 is used for a specified number of times whereas forever is used to repeat following code for infinite times.

**Q.19** Which command is used to change the musical instrument?

**Ans.**

Set instrument to is used to change the musical instrument.

**Q.20** Which command is used to change the size of the sprite?

**Ans.**

Set size to command is used to change the size of the sprite.

**Q.21** Identify the commands with the help of the purpose given below.

1. Command is used to trigger the following code when specific key is pressed.
2. Command which is used to turn sprite to specific degrees.
3. Command which is used to set the color of the pen.

**Ans.** Following are the commands identified against above descriptions:

1. When key pressed
2. Turn degrees
3. Set pen color to

**Q.22** Create a script in scratch using following instructions.

- Create a script to glide the sprite along the sides of a triangle. The first vertex of the triangle is (-100, -100). The second vertex is (200, -100). The third vertex is (50, 100). Make sure you complete the triangle.
- Change the speed of gliding and run again.

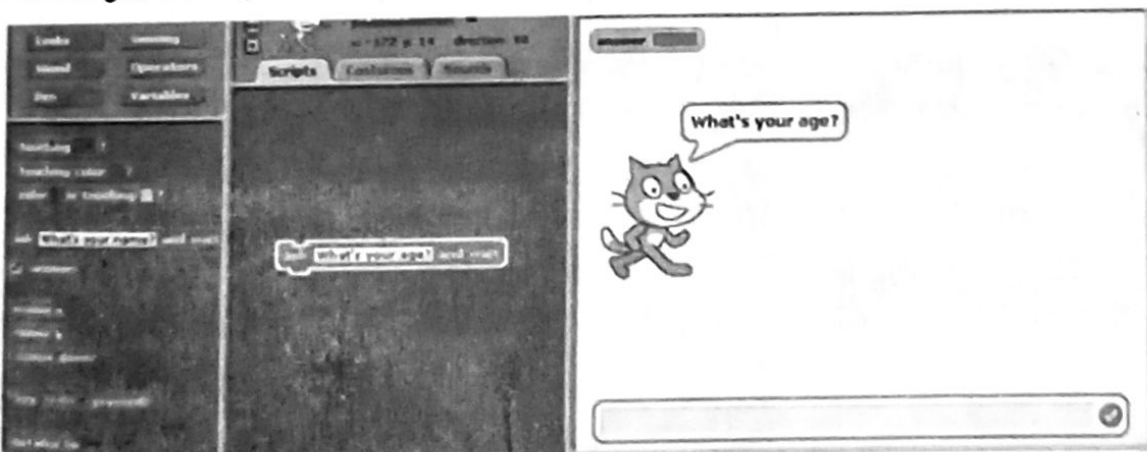
**Ans.** Following is the script for the above description:

- Start at ( -100, 100 )
- Move 200 steps
- Turn right 90 degrees
- Move 200 steps
- Turn right 90 degrees
- Move 200 steps
- Turn right 90 degrees
- Move 200 steps

**Q.23** Write a script based on the following instructions?

- Asking the age of the user
- Program wants age of user (maybe to set a level of difficulty in a game)
- Program asks user for age
- User types in age
- Program stores the answer in a variable named "answer"

**Ans.** Following is the required script:





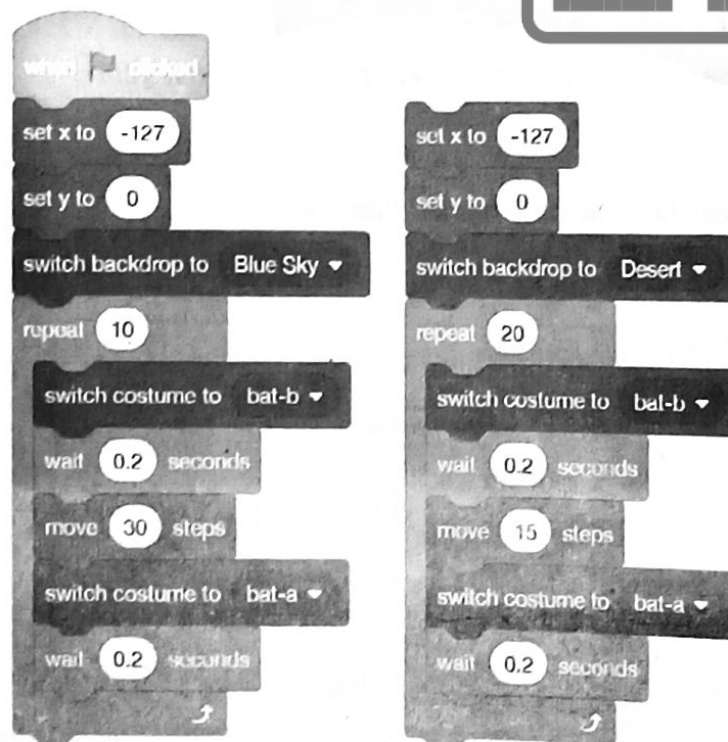
**Q.24** Write down the output of the following program?



- Ans.**
- Trigger is the space key which will initiate the code
  - Output screen will show "I can play music"
  - 6 notes 55,55,57,55,72,71 will be played twice with specified beats respectively
  - Now screen will show "Happy Birthday"
  - Drum beats 1,2,3,4 will be repeated once for 0.25 beats each respectively.

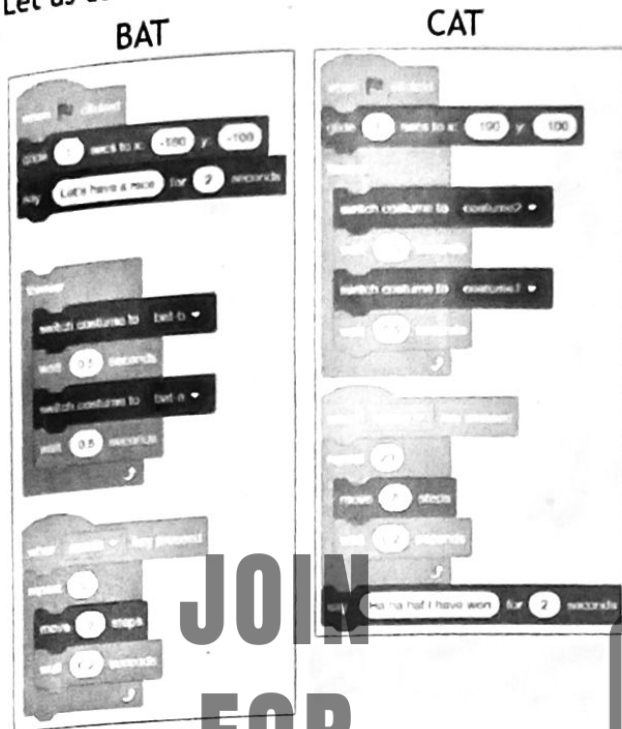
**Q.25** Create a program where dinosaur moves lazily on the screen and as soon as desert appears dinosaur becomes happy?

**Ans.** Following is the scratch code:cratch code:



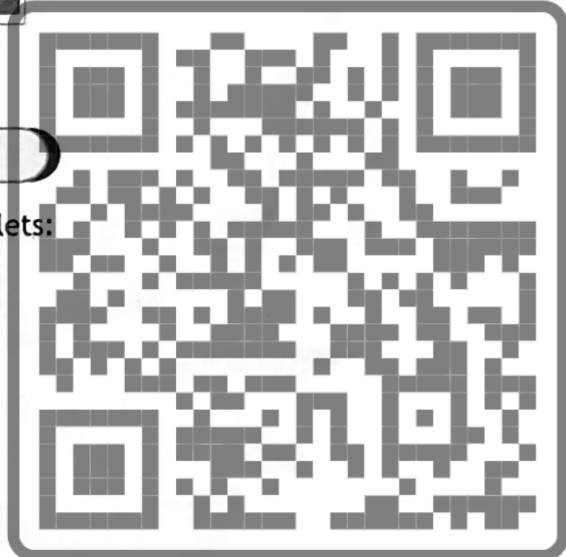
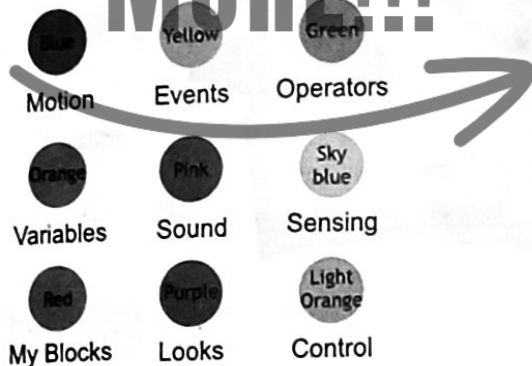
**Q.26** Write a program with multiple sprites?

**Ans.** Let us consider the default sprite cat and second sprite as bat:



**Q.27** What are the colors of the script pallets?

**Ans.** Following picture shows color of all script pallets:



**Q.28** Make list of your five favorite sprites and backdrops?

**Ans.** Five favorite sprites are:

- Microphone
- Bat
- Apple
- Dragon
- Elf

Five Favorite backdrops are:

- Arctic
- Colorful city
- Desert
- Spaceship
- Wetlan



**Q.29** Create two sprites and make them talk to each other also try to input some jokes?

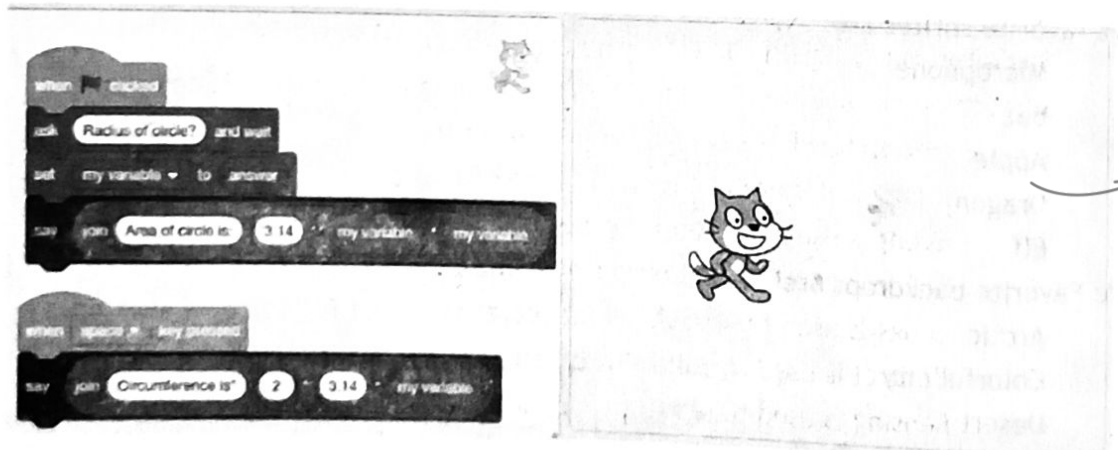
**Ans.** Following is the scratch story:



JOIN  
FOR  
MORE!!!

**Q.30** Calculate area and circumference of the circle in scratch?

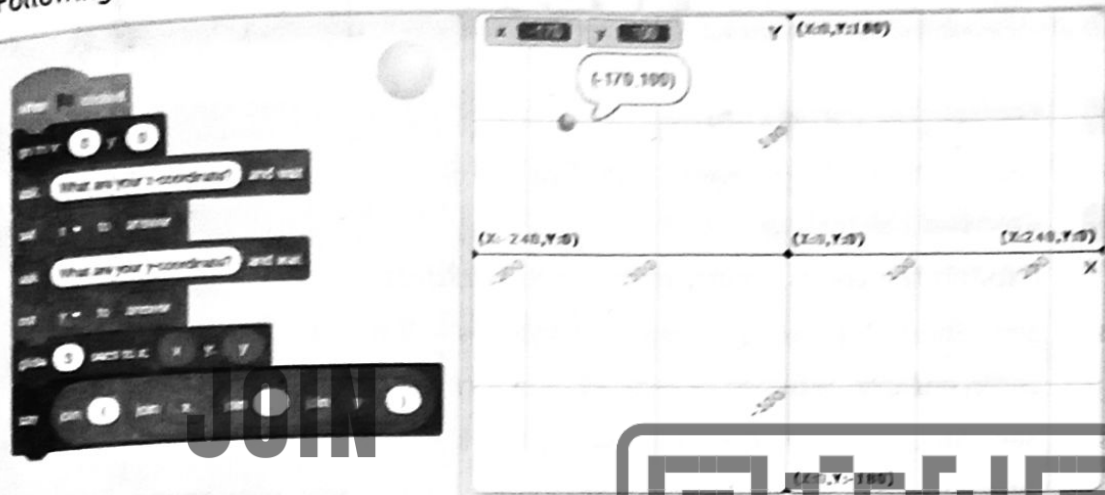
**Ans.** Following is the scratch program to find out area of the circle:



**Q.31** Write a program to mark following coordinates on a matrix.

- A.  $x=-170, y=100$
- B.  $x=110, y=190$
- C.  $x=-120, y=-120$
- D.  $x=150, y=0$

**Ans.** Following is the code:



## Summary

- Scratch is a programming language. It includes animations and games.
- We can create interactive stories, games, animation, music, art, and presentations.
- There are two basic components of a Scratch program; Sprite and Script.
- Scratch Environment includes: Stage or Stage Preview Window, Script Area, Sprite List, Backdrop, and Script Block.
- We can use Scratch online and offline.
- The Stage is where you see your stories, games, and animations come to life.
- The Script is the set of stepwise instructions that users give to the sprite to do a particular task.
- Backdrop is the background that the user can add on stage.
- Scratch blocks are organized into different categories in the left columns.
  - ✓ Motion block has instructions to make the Sprite move, such as
  - ✓ number of steps to take, direction of motion, etc.,
  - ✓ Event is used to trigger code at a specific time or action.
  - ✓ Sound is used to play different sounds.
  - ✓ Looks is used to change the appearance of the Sprite and Backdrop.
  - ✓ Control is used to control the action on the stage.
  - ✓ Sensing is used to sense any specific happening.
- Backdrop is the background of stage.
- Costume is the appearance of Sprite.



- Scratch has a 2D coordinate system: "x position" and "y position".
- User can choose Backdrop or Sprite from library, Paint New, Upload from File and Capture from Camera.
- Variables can be created through Data Pallet.

## Solution of Textbook Exercise

**A.** Encircle the correct answer:

Ans. See "Multiple Choice Question (MCQS)" Q1-10

**B.** Respond the following:

1. Explain the terms script, sprite, and backdrop?

Ans. See "Short/Detailed Questions and Answers" - Q.8

2. Differentiate between repeat 10 and forever command?

Ans. See "Short/Detailed Questions and Answers" - Q.18

3. Write use of the following codes: Forever, wait, say, play sound, go to x, y

Ans. See "Short/Detailed Questions and Answers" - Q.17

4. What is the difference between scratch online and offline?

Ans. See "Short/Detailed Questions and Answers" - Q.6

5. Mark the colors of all available pallets in scripts tab?

Ans. See "Short/Detailed Questions and Answers" - Q.27

## Lab Activity

1. Draw a Matrix of 480 x 360 and mark following points:

A.  $x=-170, y=100$

B.  $x=110, y=190$

C.  $x=-120, y=-120$

D.  $x=150, y=0$

Ans. See "Short/Detailed Questions and Answers" - Q.31

2. Make list of your five favorite sprites and backdrops?

Ans. See "Short/Detailed Questions and Answers" - Q.28

3. Make a sample program in which two sprites talk to each other. Try making few jokes.

Ans. See "Short/Detailed Questions and Answers" - Q.29

4. Draw a program to enter radius of the circle: Calculate area and circumference

Ans. See "Short/Detailed Questions and Answers" - Q.30