ANSWER KEY

Touchpad MODULAR Ver 1.0

Class-10

1. Introduction to C++

EXERCISE

A. 1. (a) 2. (a) 3. (d) 4. (b) 5. (a) 6. (b)

B. 1. (T) 2. (T) 3. (T) 4. (F) 5. (T)

C. 1. curly brace 2. Run 3. pre-processor 4. semicolon 5. AT&T Bell Labs

D. 1. ISO stands for International Organization for Standardization.

2. We need to compile a program to check whether it contains any warning or error.

3. The #include<iostream.h> header file is used to contains pre-defined objects like cout<<.

E. 1. Four features of C++ language are following:

(i) Object-Oriented Programming

(ii) Platform Independent

(iii) High-Level Language

(iv) Case-Sensitive

2. Compile the program is use to check whether the program contains any warning or error. After compiling the program, we run the program, for the output.

IN THE LAB Subject Enrichment

Do yourself.

2. Getting Started with C++

EXERCISE

A. 1. (a) 2. (b) 3. (a) 4. (d) 5. (a) 6. (d) 7. (d)

B. 1. (T) 2. (T) 3. (T) 4. (F) 5. (F)

C. 1. Floating-Point 2. Initializing 3. strongly-typed

4. memory 5. variable

IMPRINT 1

D. 1. Primitive data types are built-in or default data types. For example int, float, char etc..

2. A data type specifies the type of value a variable can contain. Three basic data type used in C++ are int, float, and char.

3. A character set refers to the letters, digits or special symbols that can be used to write programs in C++ language.

4. The syntax to declare a variable in C++ is as follows:

```
<type> <variable_name>;
```

E. 1. (i) A variable name cannot start with a digit.

(ii) Keywords cannot be used as variable names.

(iii) Variable names are case sensitive.

(iv) Variable names cannot contain blank space.

2. C++ is a strongly-typed language, which means you need to declare a variable before using it.

3. C++ allows some special characters in its coding called delimiters. Operators are special symbols that are used to perform calculations.

IN THE LAB SUBJECT ENRICHMENT

Do yourself.

Periodic Assessment-1

(Based on chapters 1 & 2)

A. 1. Menu bar help user to do various tasks like create a new file, save a program, compile a program and run a program.

2. C++ supports multiple platforms which platform independent means the program developed on a Windows operating system works in a similar manner on other operating systems like Linux.

3. The main( ) function is the entry point for the compiler. Every statement of the program should be written inside the main( ) function.

4. The getch() function is used to take an input from the output window during runtime. It keeps the output window open until you press any key from the keyboard.

B.

```
#include<iostream.h>
#include<conio.h>
void main()
{
cout<<"Welcome to Orange Education"
getch();
}
```

C. 1. c. 2. d. 3. e. 4. b. 5. a.

3. Operators in C++

Exercise

A. 1. (d) 2. (c) 3. (b) 4. (d) 5. (d)

B. 1. (F) 2. (T) 3. (T) 4. (T) 5. (F)

C. 1. Modulus 2. == 3. AND 4. Assignment 5. Increment

D.
1. The difference between '/' and '%' operators is that '/' operator gives the quotient and '%' operator gives the remainder obtained during division.
2. Operator precedence determines the order in which the operators are executed.
3. Type casting means to convert the value of one data type into another data type.
4. Relational operators are used to compare the value of the two operands and return True or False accordingly.
5. The operators work on two operands are called binary operators.

E.
1. C++ provides the following relation operators: Equal to, Not equal to, Greater than, Less than, Greater or equal to, Less or equal to.
2. In case of prefix operators the value of the variable is incremented/decremented first and then the expression is evaluated whereas in case of postfix operators, the expression is evaluated first and then the value of the variable is incremented/ decremented.
3. Implicit type casting is also known as automatic type casting. It is done by the C++ compiler while compiling the program. Explicit type casting means to convert one data type into another using typecast operator.
4. Both are assignment operator. *= means it multiplies right operand with the left assignment operand and assigns the result to left operand. /= means it divides left operand with the right operand and assigns the result to left operand. x/=3 is equivalent to x=x/3.

F. 1. 88 2. 4 3. Grade is: A

4. (This question was printed incorrectly in the book. Please correct it in you text book.)

Q.
```
void main()
    {
    int x , y = 3, z = 6;
    x = ++z + ++y / z;
    cout<<"value of x is: "<<x;
   }
```

Ans. value of x is: 7

In the Lab
Subject Enrichment

Do yourself.

4. Input and Output in C++

Exercise

A. 1. (a) 2. (a) 3. (d) 4. (a) 5. (c)

B. 1. (T) 2. (T) 3. (F) 4. (T)

C. 1. comments 2. cin 3. ostream 4. break

D. 1. There are two types of comments used in C++. Single line comment and Multiple line comment.

2. \\ (two backward slashes) symbol is used to add backslash also known as escape sequence characters and // (two forward slashes) symbol is known as a comment.

3. Escape sequence characters are used to represent certain special characters which cannot print directly on the output screen.

E. 1. The cout object is used to display the output on the output device connected to the computer like monitor. For example: cout<<"Output on the screen" ;

2. The cout object is used to display the output on the output device connected to the computer like monitor whereas the cin object is used to take the input from the user through keyboard.

F. 1. Hello Touchpad

2. Enter your age: 19

19

In the Lab

Subject Enrichment

Do yourself.

5. Conditional Statements

Exercise

A. 1. (d) 2. (b) 3. (a) 4. (c) 5. (a)

B. 1. (F) 2. (F) 3. (F) 4. (T) 5. (T)

C. 1. if 2. true 3. false 4. else 5. break

D. 1. Conditional statements are used to test the conditions and decide the flow of a program on the basis of the results of the conditions.

2. The if...else statement is used to evaluates one block of statements out of two.

3. Yes

4.
```
if ( condition1 ) {
// statement 1
```

```
if (condition2 ) {
// statement 2
. . . and so on
}
}
```

E. 1. The if statement selects and executes the statement(s) based on a given condition, If a particular condition or expression evaluates to **True**, then a set of statements will be executed. Otherwise, the control of execution is passed to the next statement after the if block.

```
if (condition) {
statement(s) ;
}
```

2. Switch statement uses case statement multiple times to evaluate multiple expressions. If statement can test any type of expressions (integer, floating-point, logical, relational, etc.), while switch can test only integer expressions. break can be used inside switch body but cannot be used inside if body.
3. Switch statement uses case statement multiple times to evaluate multiple expressions. If none of the expressions evaluates to true, then the default case gets executed.
4.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    char code;
    cout<<"enter any character from (A, C, G, F)";
    cin>>code;
switch (code)
{
case 'A' : cout<<"\n \t Accountant";
break;
case 'C' : cout<<"\n \t Grade IV";
break;
case 'G' : cout<<"\n \t Grade IV";
break;
case 'F' : cout<<"\n \t Financial Advisor";
break;
default :
cout<<"\n \t Wrong character entered";
}
getch();
}
```

5.
```
#include<iostream.h>
#include<conio.h>
void main()
 {
   char draw;
cout<<"Enter any character from (R, C, E, P)";
cin>>draw;
switch (draw) {
case 'R' :
cout << "Draw rectangle";
break;
case 'C' : cout << "Draw circle";
break;
case 'E' : cout << "Draw ellipse";
break;
case 'P' : cout << "Draw polygon";
break;
default :
cout << "Wrong choice";
}
   getch();
}
```

6.
```
#include<iostream.h>
#include<conio.h>
void main()
{
int time = 08;
if(time <12 )
{
cout<<"Good morning.";}
else if ( time > 20){
cout<<"Good night.";}
else {
cout<<"Good evening.";
}
getch();
}
```

IN THE LAB SUBJECT ENRICHMENT

Do yourself.

Periodic Assessment-2

(Based on chapters 3 to 5)

A. 1.

Not equal to operator	checks if the values of two operands are not equal and returns false if both are equal.	x != y	TRUE

2.

AND operator	returns true, if both operands are true.	(x < 5) && (x < 10)	TRUE

3.

Remainder assignment	takes modulus of two operands and assigns the result to left operand. x%=3 is equivalent to x=x%3.	x %= 3

4. Increment (++) operator which is used to increase the value of a variable by 1. It is also known as a unary operators. For example:

```
a = ++a;          means a = a + 1
```

B. Good evening.

C. Do yourself.

Test Sheet-1

(Based on chapters 1 to 5)

Section A

A. 1. (a) 2. (d) 3. (d) 4. (d) 5. (a) 6. (a)
7. (d) 8. (d) 9. (a) 10. (c)

B. 1. semi-colon 2. Memory 3. ==
4. iostream 5. Curly brackets 6. Case

C. 1. (T) 2. (T) 3. (T) 4. (T) 5. (F) 6. (T)

Section B

A. 1. The #include<iostream.h> header file is used to contains pre-defined objects like cout<<.

2. The syntax to declare a variable in C++ is as follows:

```
<type> <variable_name>;
```

3. The difference between '/' and '%' operators is that '/' operator gives the quotient and '%' operator gives the remainder obtained during division.

4. \\ is used to add backslash also known as escape sequence characters and // (two forward slashes) symbol is known as a comment.
5. The if ... else statement checks for a condition. If the condition evaluates to True, the statement(s) in the if block is executed, otherwise the statements in the else block is executed.
6. Yes we can.

B. 1. (i) A variable name cannot start with a digit.
(ii) Keywords cannot be used as variable names.
(iii) Variable names are case sensitive.
(iv) Variable names cannot contain blank space.

2. Both are assignment operator. *= means it multiplies right operand with the left assignment operand and assigns the result to left operand. /= means it divides left operand with the right operand and assigns the result to left operand. x/=3 is equivalent to x=x/3.
3. Compile the program is use to check whether the program contains any warning or error. After compiling the program, we run the program, for the output.
4. The cout object is used to display the output on the output device connected to the computer like monitor whereas the cin object is used to take the input from the user through keyboard.
5.
```
#include<iostream.h>
#include<conio.h>
void main()
{
   int number;
cout<<" Enter any number from 1-7";
cin>>number;
switch (number)
{
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 << "Wrong choice";
}
getch();
}
```

6. Switch statement uses case statement multiple times to evaluate multiple expressions. If none of the expressions evaluates to true, then the default case gets executed.

6. Loops

EXERCISE

A. 1. (d) 2. (b) 3. (a) 4. (a) 5. (c)

B. 1. (F) 2. (F) 3. (F) 4. (T) 5. (T)

C. 1. a++ 2. do-while 3. Iteration 4. nested 5. infinite

D. 1. The main purpose of looping is to repeat a section of your program for a certain number of times until a condition is true.

2. C++ offers four jump statements - break, continue, return and goto - which are used within the loop.

3. C++ allows you to declare and initialize a variable inside the for loop. For example:

```
for (int i = 1, j = 5 ; i <= 10; i++)
```

4. The goto statement is used to change the normal flow of program execution and transfer the control to some other part of the program.

E. 1. The break statement in C + + is used to stop the execution of a program or part of a program. The continue statement is used to skip the current iteration of the loop.

2. (i) while is pre-test (or top-test) loop. First the condition is tested, then only entry is possible in loop body. But do-while is post-test (bottom-test) loop. Hence, the loop is executed at least once.

(ii) The minimum number of iterations of a while loop is zero. But, in case of a do-while loop, the minimum number of iterations is one.

3. Yes, a while loop contain another while loop that is known as nested while loop.

```
The syntax is following:
while(condition1)
            {
```

```
        statement(s)
            while(condition2)
                                {
                                  statement(s)
                                }
    }
```

F. 1. value is not five

2. 124

3. (This question was printed incorrectly in the book. Please correct it in you text book.)

Q.
```
int sum = 0;
for (int i = 0; i <= 10; i++)
sum = sum + i;
cout <<sum;
```

Ans. 55

4.
```
* * *
* * *
* * *
* * *
```

IN THE LAB Subject Enrichment

Do yourself.

7. OOP Concepts

Exercise

A. 1. (d) 2. (a) 3. (a) 4. (d) 5. (a) 6. (b)

B. 1. (T) 2. (T) 3. (F) 4. (T)

C. 1. Object-oriented programming is a programming paradigm that focuses on objects instead of routines or functions.

2 Encapsulation refers to a process of binding data and function together into a single unit like a capsule.

3. Inheritance allows a class to acquire or inherit all the properties and behaviors from its parent class.

4. Function overloading is the process of creating more than one function with same name but different numbers, type or sequence of parameters.

D. 1. Function overloading is the process of creating more than one function with same name but different numbers, type or sequence of parameters.

Function overriding means to use the same function with same name, return-type and arguments list in both sub class and super class.

2. A class can be defined as a user defined blueprint or prototype that is used to create objects. An object is a real-world entity like car, dog, pencil and computer. All the real-world entities have properties and behavior.

3. The class which inherits the members of another class is called derived class or sub class. On the other hand, the class whose members are inherited is called base class or super class.

IN THE LAB SUBJECT ENRICHMENT

Do yourself.

Periodic Assessment-3

(Based on chapters 6 & 7)

A. 1 2 3 4 5 6 7 8 9 10

B. Do yourself.

C. 1. The term polymorphism is made up of two words "poly" and "morph" which means many forms.

2. An object is a real-world entity like car, dog, pencil and computer. All the real-world entities have properties and behaviour. Properties represent the physical appearance and qualities of an object.

3. Encapsulation refers to a process of binding data and function together into a single unit like capsule.

8. Functions in C++

EXERCISE

A. 1. (a) 2. (a) 3. (b) 4. (a) 5. (a) 6. (a)

B. 1. (T) 2. (T) 3. (T) 4. (F) 5. (F)

C. 1. scope resolution 2. reference 3. arguments 4. definition 5. functions

D. 1. A function is a block of organized and reusable code used to perform a particular task.

It allows you to reuse a code many times and reduces duplicity in a program.

2. Before using a function, we need to declare it. Declaring a function is called function prototyping. A function prototype tells the compiler about the data type and parameters of the function.

3. In function definition, if you do not want to return any value from the function, then you need to declare a function with the void data type. In this case, the function returns 0 (zero).

4. The return is jump statement in C++ that transfer the controls to the calling function. The default return type of every function in C++ is int.
5. The variables that are declared outside of all the functions are called global variables. Global variables can be used by any function in the program.

E. 1. Declaring a function is called function prototyping. A function prototype tells the compiler about the data type and parameters of the function. After declaring a function, you need to define it. Defining a function means to give the body of the function.

Example to declare a function named display:

```
int display(int);
```

the example of display function:

```
int display(int age)
```

2. The variables which are declared in the body of a function or block are called local variables for that function. The variables that are declared outside of all the functions are called global variables
3. The parameters passed at the time of declaring a function are called formal parameters. The parameters passed at the time of calling a function are called actual arguments.
4.
```
void name()
{
int i;
for( i = 1; i <= 5; i++ )
{
cout<<"\n"<<" REEMA";
}
}
```

F. 1.
```
int div(int a, int b)
{
a = a * b;
b = b / 2;
return(a,b);
}
```

2.
```
int sum(int i, j)
{
return (i / j);
}
```

3.
```
void disp() {
{ cout<<”program”;
}
```

IN THE LAB Subject Enrichment

Do yourself.

9. Header Files and Library Functions

EXERCISE

A. 1. (a) 2. (a) 3. (a) 4. (a) 5. (a)

B. 1. (F) 2. (T) 3. (T) 4. (T) 5. (T)

C. 1. header 2. log() 3. floor( ) 4. extension 5. isdigit( )

D. 1. A header file can be defined as a collection of predefined functions which you can use to increase your program's productivity and save your programming time.

2. The three functions provided by math.h header file are following:

   (i) floor(a) (ii) pow(a, b) (iii) sqrt(a)

3. You can include a header file in your program by using the preprocessor directive #include followed by name of the header file in between the angular brackets:

   #include<iostream.h>

E. 1. isalpha(c) returns true if the character c is an uppercase or a lowercase letter. Otherwise returns false and isalnum(c) returns true if character c is a digit from 0 through 9 or a letter either uppercase or lowercase. Otherwise returns false.

2. ceil(a) returns a after rounding up it to the nearest integer and floor(a) returns a after rounding down it to the nearest integer.

IN THE LAB Subject Enrichment

Do yourself.

10. App Development

EXERCISE

A. 1. (a) 2. (a) 3. (a) 4. (a) 5. (b) 6. (a)
7. (a) 8. (d)

B. 1. (F) 2. (T) 3. (F) 4. (T) 5. (T) 6. (F)

C. 1. hybrid 2. Mobile 3. Android 4. gaming 5. install

D. 1. An app is a software program primarily developed for hand-held smart devices such as mobile and tablet.

2. Web apps are actually web applications which give a user experience similar to native apps.
3. Native apps are the type of Mobile apps. These are platform dependent which means that these apps are primarily developed for a specific platform.
4. E-commerce apps are the apps that allow us to buy or sell products while sitting at home or any other place.

E. 1. (i) Gaming Apps: Today's most popular category of mobile apps is gaming apps which share more than 24% area of the app store. The most commonly used gaming apps are PUBG, Candy Crush Saga, and Angry Birds.

(ii) Productivity apps, also known as business apps used by businessmen to perform several complex tasks. The most commonly used productive apps are Google Calendar, Evernote and Dropbox.

(iii) Entertainment apps are developed to entertain the people. The most commonly used entertainment apps are Netflix, Talking Tom and YouTube.

2. Two categories of mobile apps are Native Apps and Web Apps.

Native apps are platform dependent which means that these apps are primarily developed for a specific platform. For example, any app which is developed for iOS will not run on any other platform such as Android, Windows, and BlackBerry.

Web apps are actually web applications which give a user with experience similar to native apps. Some of the examples of the Web apps are OLX, Flipkart, and Pinterest.

3. The difference between web app and website is that a web app can be a small part of a website which provides a particular functionality. On the other hand, a website can contain many web apps.
4. Educational apps provide a platform for children to learn from anywhere and anytime. These apps use advance methodologies and new concepts to make the learning easier. The most commonly used educational apps are Khan Academy, Vedantu, and Grammar EN.

IN THE LAB Subject Enrichment

Do yourself.

Periodic Assessment-4

(Based on chapters 8 to 10)

A.

1. An App	2. Play Store	3. iOS	4. Web Applications
5. Apple Play Store	6. Hybrid Apps	7. Gaming Apps	8. Educational Apps
9. Social Networking Apps		10. Web Apps	

B. 1. True 2. True 3. False 4. False 5. False

C. Do yourself.

Test Sheet-2

(Based on chapters 6 to 10)

Section A

A. 1. (a) 2. (a) 3. (a) 4. (a) 5. (c) 6. (a)
7. (d) 8. (b) 9. (d) 10. (a)

B. 1. isdigit( ) 2. do while 3. else 4. reference 5. functions 6. header file
7. Android

C. 1. (F) 2. (T) 3. (T) 4. (T) 5. (T) 6. (F)
7. (F)

Section B

A. 1. The return is jump statement in C++ that transfer the controls to the calling function. The default return type of every function in C++ is int.

2. Inheritance allows a class to acquire or inherit all the properties and behaviors from its parent class.

3. C++ offers four jump statements - break, continue, return and goto, which are used within the loop.

4. You can include a header file in your program by using the preprocessor directive #include followed by name of the header file in between the angular brackets:

```
#include<iostream.h>
```

5. C++ allows you to declare and initialize a variable inside the for loop. For example:

```
for (int i = 1, j = 5 ; i <= 10; i++)
```

6. E-commerce apps are the apps that allow us to buy or sell products while sitting at home or any other place.

B. 1. Function overloading is the process of creating more than one function with same name but different numbers, type or sequence of parameters. function overriding means to use the same function with same name, return-type and arguments list in both sub class and super class.

2. Yes, a while loop contains another while loop that is known as nested while loop.

The syntax is following:

```
while(condition1)
            {
              statement(s)
                  while(condition2)
```

```
                                    {
                                      statement(s)
                                    }
}
```

3. The parameters passed at the time of declaring a function are called formal parameters. The parameters passed at the time of calling a function are called actual arguments.
4. isalpha(c) returns true if the character c is an uppercase or a lowercase letter. Otherwise returns false and isalnum(c) returns true if character c is a digit from 0 through 9 or a letter either uppercase or lowercase. Otherwise returns false.
5. Educational apps provide a platform for children to learn from anywhere and anytime. These apps use advance methodologies and new concepts to make the learning easier. The most commonly used educational apps are Khan Academy, Vedantu, and Grammar EN.