Name : Shabir Ahmad

ID : 15226

Subject : Programming fundamental

Q = 1 (a)

## ANSWER :-

## ELSE .. IF STATEMENT :-

The else .. if statement is useful when you need to check multiple conditions within the program, nesting of if else blocks can be avoided using else .. if statement.

## SYNTAX OF ELSE .. IF STATEMENT :-

```
if (condition1)
{
    // These statements would execute if the condition1 is true
}

else .. if (condition 2)
{
    // These statements would excute if the condition 2 is true.
}

else .. if (condition 3)
{
    // These statements would excute if the condition 3 is true
}
```

ID.15226

```
.
.
else
{
  // These statements would excute if
all the conditions return false.
}
```

Q# 1 (b)

```
1   # include <iostream>
2   Using namespace std;
3   /* run this program using the console pauser or
      add your own getch.
4
5   main () {
6         int a = 10;
7         int b = 6;
8           if (a>b)
9
10    {
11        cout << "a is the largest number";
12    }
13
14    else
15    {
16        Coot << " smallest";
17    }
18
19  }
```

Q # 02

ANSWER:-

## LOGICAL OPERATORS:-

A logical operatoss is a symbal or word used to connet two or more expressions such that the value of the compound express-ion produced depends only on that of the original expressions and on the meaning of the operator. Common logical operatoss include AND, OR, and NOT.

- && (logical AND)
  - used to combine two conditions
  - true if both conditions are true

```
if (gender == 1 && age >= 65)
        Senior++;
```

- || (logical OR)
  - true if either of condition is true

```
if (semesterAvg >= 90 || finalExam >=90)
    cout << ("student grade is "A");
```

Q # 02 (b)

ANSWER:-

```
1  #include <iostream>
2  #include<conio.h>
3  using namespace std;
4  main()
```

#04

ID. 15226

```
5   {
6       int tmp;
7
8       Cout << "temperature is \n";
9       Cin >> tmp;
10      if (tmp>=40)
11      {
12        Cout "its very hot \n";
13      }
14        else if (tmp>35 && tmp<40)
15      {
16        Cout << "its tolerable \n";
17      }
18        else if (tmp>=30 && tmp<=35)
19      {
20            cout << "its warm\n";
21      }
22        else
23      {
24          Cout << " cool ";
25      }
26
27  }
```

ID.15226 #05

Q # 3 (a)

ANSWER:

A loop is used for executing a block of statements repeatedly until a particular condition is satisfied. For example, when you are displaying number from 1 to 100 you may want set the value of a variable to 1 and display it 100 times, increasing its value by 1 on each loop iteration. In C++ we have three types of basic loops: for, while and do-while. In this tutorial we will learn how to use "for loop" in C++.

**For loop.**

For loop is used to excute a set of statement repeatedly until a particular condition is satisfied. we can say it an open ended loop. General format is, for (initialization; condition; increment/decrement) { statement-block; } in for loop we have exactly two semicolons, one after initialization and second after condition In this loop we can have more than one initialization or increment/decrement, separated using comma operator. For loop can have only one condition.

**While loop.**

while loop can be address as an

entry control loop. It is completed in 3 steps.
Variable initialization. (e.g int x = 0;)

Condition (e.g while (x<=10))

Variable increment or decrement (x++ or x -- or x=x+2)

Syntax:
variable initialization; while (condition)
{statements; variable increment or decrment;}
do....while loop

In some situations it is necessary to excute body of the loop before testing the condition. Such situations can be handled with the help of do-while loop. do statement evaluates the body of the loop first and at the end, the conditions is checked
using while statement. General format of do-while loop is,
do { // a couple of statements}
while (condition);

Q # 3 (b)

ANSWER

```
1 #include <iostream>
```

```
# include <iostream>
int main () {
    using namespace std;
    int n;
    for (n = 1; n <= 5; n++){
        cout << "*" << endl;
        if (n == 2) {
            break;
        }
    }
```

## CONTINUE:

Continue statements works similar to break statement. The only difference is that break statement terminates the loop whereas continue statement passes control to the conditional test i.e., where the condition is checked.

In short, it passes control to the nearest conditional test in do...while loop or the condition of while in while loop, or the condition of for in for statements skipping the rest of the statements in the loop.

```
# include <iostream>
int main (){
    using namespace std;
    int n = 1;
    while (n < 10) {
        if (n == 5){
        n = n + 1;
            continue
        }
        cout << "n = " << n << endl;
        n++;
    }
    return
}
```

ID · 15226 #09

Q # 04 (b)

ANSWER:-

```
1   # include <iostream>
2   # include <conio.h>
3   using namespace std;
4   int main ()
5   {
6       int i;
7       int sum=0;
8
9   cout << "The first 10 natural numbers are:\n";
10  for (i = 1; i <= 10; i++)
11  {
12    cout << i >> " ";
13    sum= sum+i;
14  }
15  cout << "\nThe sum of first 10 natural
    numbers is\n".
16  cout<<sum;
17  return 0;
18 }
```

Q # 05

ANSWER:-

A relational operator is used to check the relationship between two operands. For example,

```
// Checks if a is greater than b
a > b;
```

Here, [>] is a relational operator. it checks if [a] is greater than [b] or not.

if the relation is true, it returns 1 whereas if the relation is false, it returns 0.

## C++ Character Set :-

In C++ character set is a set of all valid characters that can be used in a C++ program. Characters set is used to specify the characters or symbols recognized by the language. Character set is a set of all valid character that can be used for the source program text, while the execution character set is the set of characters used during the execution of the program. It is not necessary that source character set match and execution character set are same.

C++ character set, includes following characters -

Letters :- A-Z, a-z

The 26 lowercase Roman Characters:

a b c d e f g h i j k l m n o p q r s t

ID. 15226 #11

The 26 uppercase Roman Characters:
A B C D E F G H I J K L M N O P Q R S T

## C++ Constants.

Constants refers to immutable values. Constant are basically literals whose values cannot be modified or changed during the execution of program. Constant are also called as Literals. Constant must be initialized when declared as values cannot be assigned it to later. In C++, constants can be of following four basic types-

## C++ keywords:

In C++, there are is a set of reserved words that you cannot use as an identifire. These words are known as "reserved words" or "Keywords". Keywords are standard identifiers and their functions is prede--fined by the Compiler. we cannot use keywords as variable names, Class names, or method names, or as any other iden--tifier.