Name: Rizwan khan
ID #17015
Submitted to Sir Fazal-e- Malik
3rd Semester
BS CS

Question 1(a)

If statement:

The if statement is used to execute (or ignore) a set of statements after testing condition.

Purpose:

if statement evaluate condition. if condition is true it executed. If condition is false the condition is ignored and transfer to next.

Forms of if statement:

(1) Using a if statement can execute 1 condition at a time

e.g
```
if (40 > 2) {
cout << "40 is greater than 2";
```

(ii) **If - Else statement:**

Using if else statement we use more than one condition at a time

e.g

```
int n;
cout<<" Enter an integar value ? ";
cin >> n
    if (n>> 100)
    cout << " number is greater than
                100 ";
    else
        cout << " number is less than 100";
}.
```

(b) **Program:**

```
#include <iostream.h>
 main( ) {

int x, y;
cout << " Enter First value = ";
cin >> x;
cout << " Enter 2nd value = ";
cin >> y;
if ( a > b)
cout " First value is greater ";
if ( b > a )
cout " 2nd value is greater ";
}
```

Question 2

What are logical operators?

The logical operators are used to combine relational ~~operators~~ expressions or relational condition. The expression ~~(and)~~ containing logical operators is called logical expression. It is also called compound condition.

i) && → AND operator
ii) || → OR Operator
iii) ! → NOT operator.

&& AND Operator:

It is used to combine two or more relational expressions if all relational expressions are true then output returned by the compound expression is true. If not so false.

e.g $x=10, y=15, z=5$

$(x<y) \&\& (z==5)$ true.

$(x>y \&\& (z==5)$ false.

|| (OR) Operator:

It is used to combine more than one relational expression. If any on of given relation of the given is true, the output will be true otherwise the output will be false.

e.g If x=0, y=15, z=5 then

(x>y) || (z==5) true because (z==5) is true.

NOT operator (!)

The NOT operator is also known as unary operator which inverts true into false and false into true.

e.g x=5 and y=10

!(x>y) return true because (x>y) is false and the (!) NOT operator inverts the result into true.

(b) Program:

```
# include <iostream>
  int main()
{
  float temp;
cout " Enter temperature in F: ";
cin temp;
 if (temp > 40) {
cout "   very Hot ";

else if (temp > 35 && temp < 40)
{
  cout << "  Tolerable ";
}
  else if ( temp > 30 && temp < 35)
{
  cout << "  Warm ";
}
  Else if (temp < 30)
{
  cout "   cool   ";
}
  return 0;
  Getch ();
}
```

Question 3

Looping:

A statement or a set of statements that is executed repeatedly is called loop. The statement in a loop executed for a specified number of times or until some given condition remains true.

While Loop

It is a conditional loop statement. It is used to executed a statement or a set of statements as long as given condition remains true.

Syntax

```
While (condition)
    statement;
```

Example:

```
# include <iostream.h>
  main ()
{
   int a,n
   a=0;
   n=1
   while ( n<=10)
     {
      a = a+n;
     cout << n<< endl;
      n=n+2;
     }
   cout "<<" sum =  " <<s;
    }
```

(ii) The do-while loop:

The "do-while" loop is also a conditional loop statement it is like a while loop the condition is tested after executing the statement of loop.

Syntax:

```
do {
  Statements;
}
while (condition);
```

Example:

```
#include <iostream.h>
  main()
{
  int n;
   n=1
do
 {
   cout <<n<<endl;
    n++;
 }
  while ( n<=10);
  }
```

For loop:

A for loop is a repetition control structure that allows us to efficiently write a loop that needs to execute a specific number of times.

Syntax:

```
for ( initial; condition; increment)
{ statements
}
```

Example:

```
#include <iostream.h>
int main()
{
   int a ;
   for ( int a = 10; a<20; a=a++)
   {
     cout << "value of a :" << a<< endl;
}
```

(b) **Program:**

```
#include <iostream>
int main()
{
   int num, remainder;
cout " Enter number: ";
cin >> num;
Remainder = num % 2;
 if (remainder == 0)
{
cout << " num [redacted] << " Even " [redacted] << endl;

 Else
      cout << num. << " odd " << endl;
}
 return 0;
getch();
```

Question 4 (a)

Break statement:

The break statement terminates the execution of the loop when it is used inside body of loop.

The break statement terminates the loop during execution. The other statement of the loop that comes under the break statement are not executed.

Example:

```
For ( i=0 ; i<10 ; i++)
{
  if (i==4) {
     break;
}
cout << i<< endl;
}
```

Continue statement:

The continue statement breaks one iteration in the loop if a specified condition occur, and continue with next iteration in the loop.

Example:

```
For (i=0; i<10; i++)
{
 if (i==4){
  continue;
 }
 cout <<i<<endl;;
}.
```

b) **Program**

```
#include <iostream>
 int main()
{
 int sum = 0;
  For (i=1; i<=10; i++){
   Sum=Sum+i;
  }
 Cout << " Total Sum = " <<Sum;
  return 0;
```

Question 5

Array:

An Array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier.

e.g. we have five values of type int and it can be declared as an array without having declare 5 different variables.

One dimensional Array:

A One dimensional Array is a group of elements having the same data type and same name individual elements are referred to using common name and unique index of the element. The simplest form of an array.

Applying:

```
#include <iostream.h>
#include <conio.h>
 void main()
 {
  clrscr();
  int arr[5]={1,2,3,4,5};
  int i;
  for(i=0;i<5;i++)
 {
  cout<<"arr["<<i<<"]="<<arr[i]<<"\n
 }
 getch();
```

Two dimensional:

A two dimensional array can be think as a table, which will have x num of rows and y num of column.

e.g

we have 2 dimensional array which contains 3 rows and four column.

Initialization.

```
int a[3][4]={0,1,2,.... 11};
```