

IQRA NATIONAL UNIVERSITY
Department of Computer Science



Programming fundamentals

Name. : Ali Muqadas

ID. : 13131

Submitted to: Sir Dr. Fazl-e-Malik

Assignment: Final paper

Q1a What is the purpose of if statement? Discuss its two different forms with examples.

Ans **Statement Purpose:**

The if statement is used to check a condition and if the condition is true, we run a block of statements (called the if-block), else we process another block of statements (called the else-block). The else clause is optional.

Nested If else statement:

When an if else statement is present inside the body of another "if" or "else" then this is called nested if else.

Example of nested if else:

```
#include <stdio.h>

int main()
{
    int var1, var2;

    printf("Input the value of var1:");
    scanf("%d", &var1);
    printf("Input the value of var2:");
    scanf("%d",&var2);
    if (var1 != var2)
    {
        printf("var1 is not equal to var2\n");
    }
    //Nested if else
```

```

    if (var1 > var2)
    {
        printf("var1 is greater than var2\n");
    }
    else
    {
        printf("var2 is greater than var1\n");
    }
}
else
{
    printf("var1 is equal to var2\n");
} return 0;
}

```

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.

Example of else..if statement

Lets take the same example that we have seen above while discussing nested if..else. We will rewrite the same program using else..if statements

```

#include <stdio.h>

int main()
{

    int var1, var2;

    printf("Input the value of var1:");

```

```
scanf("%d", &var1);
printf("Input the value of var2:");
scanf("%d",&var2);
if (var1 !=var2)
{
    printf("var1 is not equal to var2\n");
}
else if (var1 > var2)
{
    printf("var1 is greater than var2\n");
}
else if (var2 > var1)
{
    printf("var2 is greater than var1\n");
}
else
{
    printf("var1 is equal to var2\n");
}
return 0;
```

Q1b Write a C++ program to read two numbers from keyboard and then find the

Ans #include <iostream>

using namespace std;

```
int main()
{
int num1, num2;
cout<<"Enter first number:";
cin>>num1;
cout<<"Enter second number:";
cin>>num2;
if(num1>num2)
{
    cout<<"First number "<<num1<<" is the largest";
}
else
{
    cout<<"Second number "<<num2<<" is the largest";
}
return 0;
}
```

Output:

```
Enter first number:10
Enter second number:30
Second number 30 is the largest
```

Q2a What are the Logical Operators? Explain them

Ans Logical operator ;

A logical operator is a symbol or word used to connect two or more expressions such that the value of the compound expression produced depends only on that of the original expressions and on the meaning of the operator. Common logical operators include AND, OR, and NOT.

| Operator | Name | Description | Example |
|----------|-------------|---|----------------------------|
| && | Logical and | Returns true if both statements are true | $x < 5 \ \&\& \ x < 10$ |
| | Logical or | Returns true if one of the statements is true | $x < 5 \ \ x < 4$ |
| ! | Logical not | Reverse the result, returns false if the result is true | $!(x < 5 \ \&\& \ x < 10)$ |

Q2b Write a C++ program to get Temperature in Fahrenheit F and then find the Atmosphere according to the below rules:

If temperature F is above 40 degree Fahrenheit then display.....Very Hot.

If temperature F is between 35 & 40 degree Fahrenheit then display.....Tolerable.

If temperature F is between 30 & 35 degree Fahrenheit then display.....Warm.

If temperature F is less than 30 degree Fahrenheit then display.....Cool.

Ans

```
#include<iostream>
#include<conio.h>
using namespace std;
int main ()
{
float F;
```

```

cout<<"Enter temperature:";
cin>>F;
if(F>40)
cout<<"Very Hot";
else if(F > 35 && 40)
cout<<"Tolerable.";
else if(F > 30 && 35)
cout<<"Warm.";
else
cout<<"Cool day.";
return 0;
}

```

Q3a What does **Looping** mean? Explain different loops in C++.

Ans A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages – C++ programming language provides the following type of loops to handle looping requirements.

TYPES OF LOOP IN C++ ;

Entry Controlled loops:

In this type of **loops** the test condition is tested before entering the **loop** body

. For Loop and While Loop are entry controlled loops.

Exit Controlled Loops

In this type of **loops** the test condition is tested and reevaluated at the end of **loop** body.

Q3b Write a C++ program to read a number from keyboard and then determine whether it is *Even or Odd* number?

Ans Check Whether Number is Even or Odd using if else

```
#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;
}
```

Output

Enter an integer: 23

23 is odd.

Q4a What is the purpose of using *break and continue statements*?

Ans **BREAK STATEMENT ;**

When a **break statement** is encountered, it terminates the block and gets the control out of the switch or loop.

CONTINUE STATEMENT ;

When a **continue statement** is encountered, it gets the control to the next iteration of the loop.

Q4b Write a C++ program to find the sum of the following numbers:
1+2+3+.....+10

Ans Write a program in C++ to find the sum of first 10 natural numbers.

Sample Solution :-

C++ Code :

```
#include <iostream>

using namespace std;

int main()

{

int i,sum=0;

cout << "\n\n Find the first 10 natural numbers:\n";

cout << "-----\n";

cout << " The natural numbers are: \n";

for (i = 1; i <= 10; i++)

{

cout << i << " ";

        sum=sum+i;

}

cout << "\n The sum of first 10 natural numbers: "<<sum << endl;

}
```

Sample Output:

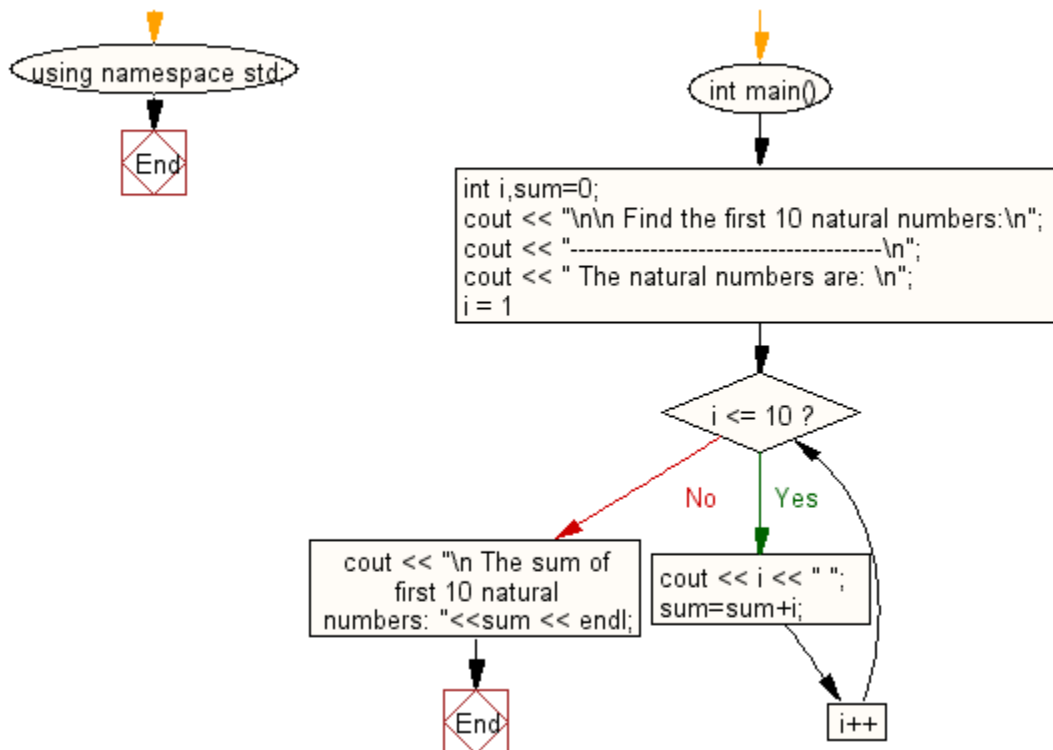
Find the first 10 natural numbers:

The natural numbers are:

1 2 3 4 5 6 7 8 9 10

The sum of first 10 natural numbers: 55

Flowchart:



Q5 What is an array? Explain On-Dimensional and Two-Dimensional Arrays with examples.

Ans ARRAY;

C++ provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a

collection of data, but it is often more useful to think of an array as a collection of variables of the same type. ... A specific element in an array is accessed by an index.

ONE DIMENSIONAL ARRAY ;

A one-dimensional array (or single dimension array) is a type of linear array. Accessing its elements involves a single subscript which can either represent a row or column index. As an example consider the C declaration `int anArrayName[10];` which declares a one-dimensional array of ten integers.

TWO DIMENSIONAL ARRAY ;

In C++ Two Dimensional array in C++ is an array that consists of more than one rows and more than one column. In 2-D array each element is refer by two indexes. Elements stored in these Arrays in the form of matrices. The first index shows a row of the matrix and the second index shows the column of the matrix.