Iqra National University Peshawar Pakistan

Department of Computer Science

Spring Semester, Final Term Exam, June 2020

Paper :	**Programming Fundamentals**	Date and Starting Time:	**26/June/2020, 9:00 am**
Program:	**BS (CS & SE)**	Uploading Date and End Time:	**26/June/2020, 3:00 pm**
Teacher Name:	**Dr. Fazal-e-Malik**	Marks	**50**
Name:	**Muhammad Hamza**	ID:	**13136**

Note: Attempt all Questions. Help can be taken from net where ever is required.

Q.1

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

Ans: The purpose of if statements are Fortran's main branching tool. They give Fortran an ability to make decisions in a program. The different forms of if statements that can be used include the simple logical if, the if-then-else structure, and the arithmetic if.

Two different forms:

There are a few different structures that must be used depending on what form of the if structure you are using. They are as follows: The simple logical if**:** In a simple logical if, all that is need is some logical expression enclosed in parenthesis and some executable statement following the logical expression. The statement will only be executed if the logical expression evaluates to. true. An example of this form of if statement is shown below.

```
if (ldata.ge.100) print *,'There are too many data',
& ' points'
```

If then - else structure:

This is the first statement in an if-then-else structure. It contains the first logical expression to be evaluated. If the logical expression evaluates to. true. then only those statements beginning with the following one and ending before the next ELSE or ENDIF will be evaluated. All other statements in the structure will be ignored and not evaluated by the compiler.

Example:

```
if ( delta.ge.1.) then
                        print *, ' Too much deflection'
                else if ( shear.ge.21.) then
                        print *, 'Beam fails in shear'
                else if ( sigma.ge.36.) then
                        print *, 'Beam fails in tension'
                else if ( sigma.le.-36.) then
                        print *, 'Beam fails in compression'
                else
                        print *, 'Beam will not fail under these'
            &   ' conditions'
                end if
```

b). Write a C++ program to read two numbers from keyboard and then find the LARGEST number of them.

- Program:

```
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

using namespace std;

int main()
{
        int num1,num2;
        cout<<"Please enter Number 1:  ";
        cin>>num1;
        cout<<endl;

        cout<<"Please enter Number 2:  ";
        cin>>num2;
        cout<<endl;

        if(num1>num2)
        {
        cout<<"The largest number is: " +num1<<endl;
        }

        Else
        cout<<"The largest number is: " +num2<<endl;

        return 0;
}
```

Q.2

a). What are the Logical Operators? Explain them

Ans: 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.

Explanations:

Within most languages, expressions that yield Boolean data type values are divided into two groups. One group uses the relational operators within their expressions and the other group uses logical operators within their expressions.

The logical operators are often used to help create a test expression that controls program flow. This type of expression is also known as a Boolean expression because they create a Boolean answer or value when evaluated. There are three common logical operators that give a Boolean value by manipulating other Boolean operands. Operator symbols and/or names vary with different programming languages.

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

- Program:

```
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

using namespace std;

int main()
{
        int temp;
        cout<<"Please enter temperature in Fahrenheit :  ";
        cin>>temp;
        cout<<endl;

if (temp>40)
{
cout<<"Very Hot"
}
if (temp<30)
{
cout<<"Cool"
}
if (temp>34 && temp<41)
{
```

```
cout<<"Tolerable"
}
if (temp>29&&temp<36)
{
cout<<"Warm"
}
          Return 0;
}
```

Q.3

a). 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 from of a loop statement in most of the programming languages. C++programming language provides the following type of loops to handle looping requirements.

1. While loop:

A while loop statement repeatedly executes a target statement as long as a given condition is true.

Syntax:

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

2. For loop:

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

Syntax:

```
for ( init; condition; increment ) {
  statement(s);
}
```

3. Do... while loop:

Unlike for and while loops, which test the loop condition at the top of the loop, the do...while loop checks its condition at the bottom of the loop.

A do...while loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

Syntax:

```
            do {
  statement(s);
}
while( condition );
```

4. Nested loop:

A loop can be nested inside of another loop. C++ allows at least 256 levels of nesting.

Syntax:

```
            for ( init; condition; increment ) {
  for ( init; condition; increment ) {
    statement(s);
  }
  statement(s); // you can put more statements.
}
```

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

- Program:

```
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

using namespace std;

int main()
{
   int num;
   cout << "Enter an Number : ";
   cin >> n;

   if ( num % 2 == 0)
{
      cout << num << " is even.";
}
   else
   {
   cout << num << " is odd.";
}
   return 0;
}
```

Q.4

a). What is the purpose of using ***break and continue statements***?

Ans:

Break statement:

In any loop break is used to jump out of loop skipping the code below it without caring about the test condition.

It interrupts the flow of the program by breaking the loop and continues the execution of code which is outside the loop.

Continue statement:

Like a break statement, continue statement is also used with if condition inside the loop to alter the flow of control.

When used in while, for or do. While loop, it skips the remaining statements in the body of that loop and performs the next iteration of the loop.

Unlike break statement, continue statement when encountered doesn't terminate the loop, rather interrupts a particular iteration.

b). Write a C++ program to find the sum of the following numbers:

1+2+3+……+10

- Program:

```
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

using namespace std;

int main()
{

int n1 =1;
int n2 =2;
int n3 =3;
int n4 =4;
int n5 =5;
int n6 =6;
int n7 =7;
int n8 =8;
int n9 =9;
int n10 =10;
 //sir I took variables instead of direct number addition just because in future we can take input from user and store in variable. TIA

int sum = n1+n2+n3+n4+n5+n6+n7+n8+n9+n10;
cout<<"The sum of 1+2+3+……+10 is: "+sum;
cout<<endl;

return 0;
}
```

Q.5 What is an array? Explain On-Dimensional and Two-Dimensional Arrays with examples.

Ans: Array: The term array in programming languages are used to store multiple values in one variable, it is quite useful traditionally we have to declare a lot of variable of store multiple value but array addressed this issue.

There are generally two different types of Array.

1. Single Dimensional Array
2. Multi-Dimensional Array

Both are explained below:

One Dimensional Array: One-Dimensional Array is an array which have single row of data stored in it.

20	30	40	50	60
A[0]	A[1]	A[2]	A[3]	A[4]

We can declare the above showed array in C++ as:

int A[5]; //A[5] is the array which have length of 5. Means we can store 5 values in this array

// Storing Values init

A[0]=20;

A[1]=30;

A[2]=40;

A[3]=50;

A[4]=60;

//Displaying Array Values:

Cout<<A[0]; // This will display the values store in A[0] index.

Two Dimensional Array: Two-Dimensional Array is an array which have two rows of data stored in it. It is also know as 2D – Array

20 - A[0][0]	30 - A[0][1]	40 - A[0][2]
50 - A[1][0]	60 - A[1][1]	60 - A[2][2]

Here is my array A[2][3] which is conceptually viewed as the figure drawn above.

We can declare the above showed 2D-array in C++ as:

int A[2][3];

//// Storing Values init

A[0][0]=20;

A[1][0]=50;

.

.

.

.

A[2][2]=60;

//Displaying Array Values:

Cout<<A[1][2]; // This will display the values store in A[1][2] index which is 50.