NAME :MIAN AHSAN
ID# :13213
SUBJECT:Programming Fundamentals
DATE:26TH/JUNE/2020

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

Answer

The if statement allows you to control if a program enters a section of code or not based on whether a given condition is true or false. One of the important functions of the if statement is that it allows the program to select an action based upon the user's input.

i. The IF statement executes or skips a sequence of statements, depending on the value of a Boolean expression.

ii. Else

If control reaches this keyword, the sequence of statements that follows it is executed. This occurs when none of the previous conditional tests returned

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

Answer

```
#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;
}
```

a) What are the Logical Operators? Explain them

Answer

C/C++ has many built-in operator types and they are classified as follows:

1. Arithmetic Operators: These are the operators used to perform arithmetic/mathematical operations on operands. Examples: (+, -, *, /, %,++,–).

2. Relational Operators: These are used for comparison of the values of two operands. For example, checking if one operand is equal to the other operand or not, an operand is greater than the other operand or not etc. Some of the relational operators are (==, >= , <=). To learn about each of these operators in details go to this link.

3. Logical Operators: Logical Operators are used to combine two or more conditions/constraints or to complement the evaluation of the original condition in consideration. The result of the operation of a logical operator is a boolean value either true or false. For example, the **logical AND** represented as **'&&' operator in C or C++** returns true when both the conditions under consideration are satisfied. Otherwise it returns false. Therfore, a && b returns true when both a and b are true (i.e. non-zero)

4. Bitwise Operators: The Bitwise operators is used to perform bit-level operations on the operands. The operators are first converted to bit-level and then the calculation is performed on the operands. The mathematical operations such as addition, subtraction, multiplication etc. can be performed at bit-level for faster processing. For example, the **bitwise AND** represented as **& operator in C or C++** takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1.

b) 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.

Answer

```
#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;
}
```

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

Answer

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

i. **While loop** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

ii. **For loop** Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

iii. **Do-while loop** Like a 'while' statement, except that it tests the condition at the end of the loop body.

iv. **Nested loops** You can use one or more loop inside any another 'while', 'for' or 'do..while' loop.

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

Answer

```
#include<iostream>
using namespace std;

int main()
{
    int number, remainder;

    cout << "Enter the number : ";
    cin >> number;
    remainder = number % 2;
    if (remainder == 0)
        cout << number << " is an even number " << endl;
    else
        cout << number << " is an odd number " << endl;

    return 0;
}
```

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

Answer

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

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

1+2+3+……..+10

Answer

```
#include <iostream>
using namespace std;
int main()
{
    int i,sum=0;
    cout << "\n\n find the sum of the following numbers\n";
    for (i = 1; i <= 10; i++)
    {
        cout << i << " ";
        sum=sum+i;
    }
     cout << "\n the sum of the following numbers is "<<sum << endl;
}
```

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

Answer

An **array** is a collection of elements of the same type placed in contiguous memory locations that can be individually referenced by using an index to a unique identifier. Five values of type int can be declared as an **array** without having to declare five different variables (each with its own identifier).

i. **One dimensional array**

A **one**-**dimensional array** is a structured collection of components (often called **array** elements) that can be accessed individually by specifying the position of a component with a single index value. ... creates the number **array** which has 50 components, each capable of holding **one** int value.

ii. **Two dimensional array**

Two-**dimensional Arrays**. ... A **2D array** has a type such as int[][] or String[][], with two pairs of square brackets. The elements of a **2D array** are arranged in rows and columns, and the new operator for **2D arrays** specifies both the number of rows and the number of columns.