**Experiment No. 1**

**Title:** **Implementation of Simple C++ program**

**Batch: B2 RollNo.:16010421119 Experiment No.: 1**

**Aim**: Write a C++ program to accept two numbers from the user and perform addition, subtraction, multiplication, division and modulus operations on the two given numbers

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Resources needed: Text Editor, C++ compiler**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

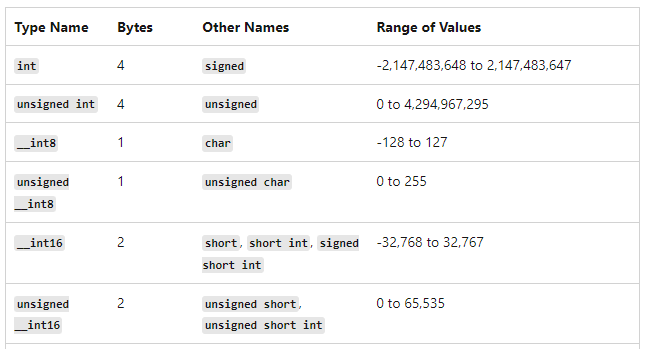
### Theory:

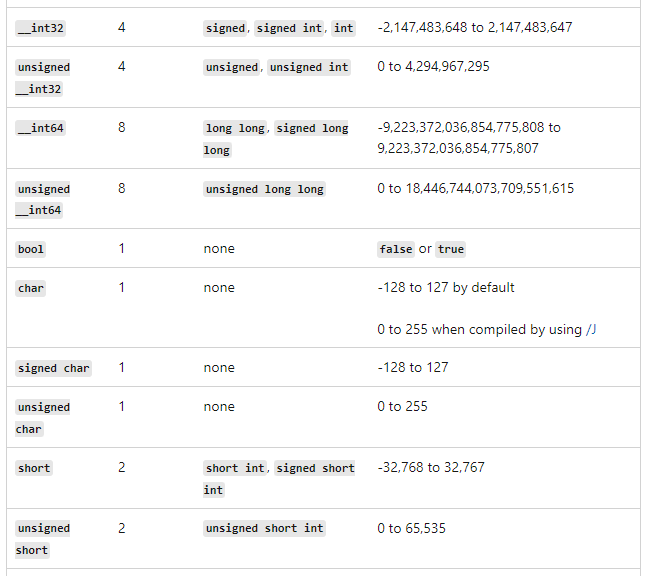
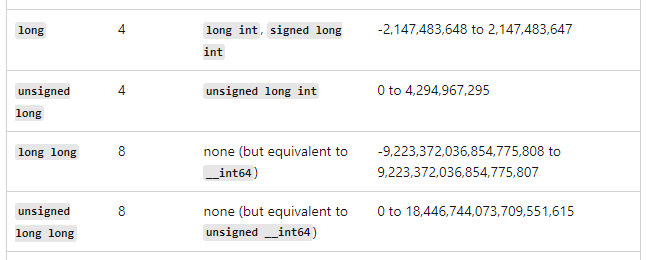
C++ is an Object-oriented programming language developed at Bell Labs by Bjarne Stroustrup. Object-oriented programming shifts the focus of attention to the objects, that is, to the aspects on which the problem is centred. OOP objects combine data (properties) and functions (capacities). A class defines a certain object type by defining both the properties and the capacities of the objects of that type. Objects communicate by sending each other “messages,” which in turn activate another object’s capacities.

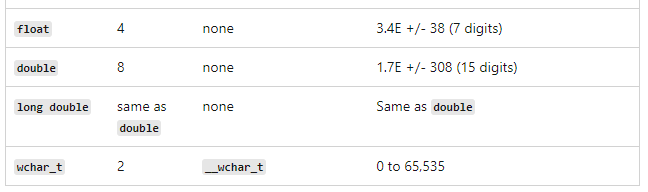
A C++ program is made up of objects with their accompanying member functions and global functions, which do not belong to any single particular class. Each function fulfils its own particular task and can also call other functions. We can have user defined functions or can use functions from standard library. There is always main( ) function defined in every C++. C++ program execution starts from main( ) function.

**Datatypes in C++**

The following table gives information about all the built-in data types of C++. the ranges specified in the table are inclusive-inclusive.







**Structure of a C++ program**

#include<iostream>

using namespace std;

int main( )

{

int a;

cout<<”Enter a number”<<endl;

cin >> a;

cout<<”Number entered by user is “<<a;

}

The structure of a sample C++ program and main ( ) function is discussed here. The first line begins with the hash symbol, #, which indicates that the line is intended for the preprocessor. The preprocessor is just one step in the first translation phase and no object code is created at this time. The header file iostream comprises conventions for input and output streams. The word stream indicates that the information involved will be treated as a flow of data. Predefined objects in C++ are defined in the std (standard) namespace. The using directive allows direct access to the names of the std namespace. Program execution begins with the first instruction in function main( ), and this is why it is mandatory that each C++ program must have a main function.

In the example the function main( ) contains input/output statements. The statement cout<<”Enter a number”<<endl; outputs the text string “Enter a number” on the screen and then because of ‘endl’ manipulator specified at the end of the cout statement, the control goes to the next line. The name cout (console output) designates an object responsible for output. The operator << is called the ‘insertion or put to’ operator. It is essentially an overloaded left-shift bitwise operator. This operator directs the contents of the variable on its right to the object on its left. Similar to cout object, for input through keyboard, cin (console input) object is used. ‘cin’ is an predefined object in C++ which corresponds to standard input stream. The ‘>>’ operator is called the ‘extraction or get from’ operator. It is essentially an overloaded right-shift bitwise operator. This operator takes the value from the stream object on its left and places it in the variable on its right. Thus is C++ input/output is performed using cin and cout objects.

**Results: (Program with snapshot of output)**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int choice;**

**int a,b,sum,diff,mult,remain;**

**double division,doublea,doubleb;**

**do**

**{**

**cout<<"Enter two numbers separated by space: ";**

**cin>>a>>b;**

**cout<<"Enter 1 for Addition of entered numbers"<<endl;**

**cout<<"Enter 2 for Subtraction of entered numbers"<<endl;**

**cout<<"Enter 3 for Multiplication of entered numbers"<<endl;**

**cout<<"Enter 4 for Division of entered numbers"<<endl;**

**cout<<"Enter 5 to Quit"<<endl;**

**cout<<"Enter choice: ";**

**cin>>choice;**

**if (choice==1)**

**{**

**sum = a+b;**

**cout<<"Addition of two numbers is: "<<sum<<endl;**

**}**

**else if(choice==2)**

**{**

**diff = a-b;**

**cout<<"Difference of two numbers is: "<<diff<<endl;**

**}**

**else if(choice==3)**

**{**

**mult = a\*b;**

**cout<<"Multiplication of two numbers is: "<<mult<<endl;**

**}**

**else if(choice==4)**

**{**

**remain = a%b;**

**doublea = static\_cast<double>(a);**

**doubleb = static\_cast<double>(b);**

**division = doublea/doubleb;**

**cout<<"Division of two numbers is: "<<division<<endl;**

**cout<<"Remainder after division of two numbers is: "<<remain;**

**}**

**else**

**{**

**cout<<"Please enter a valid input!!"<<endl;**

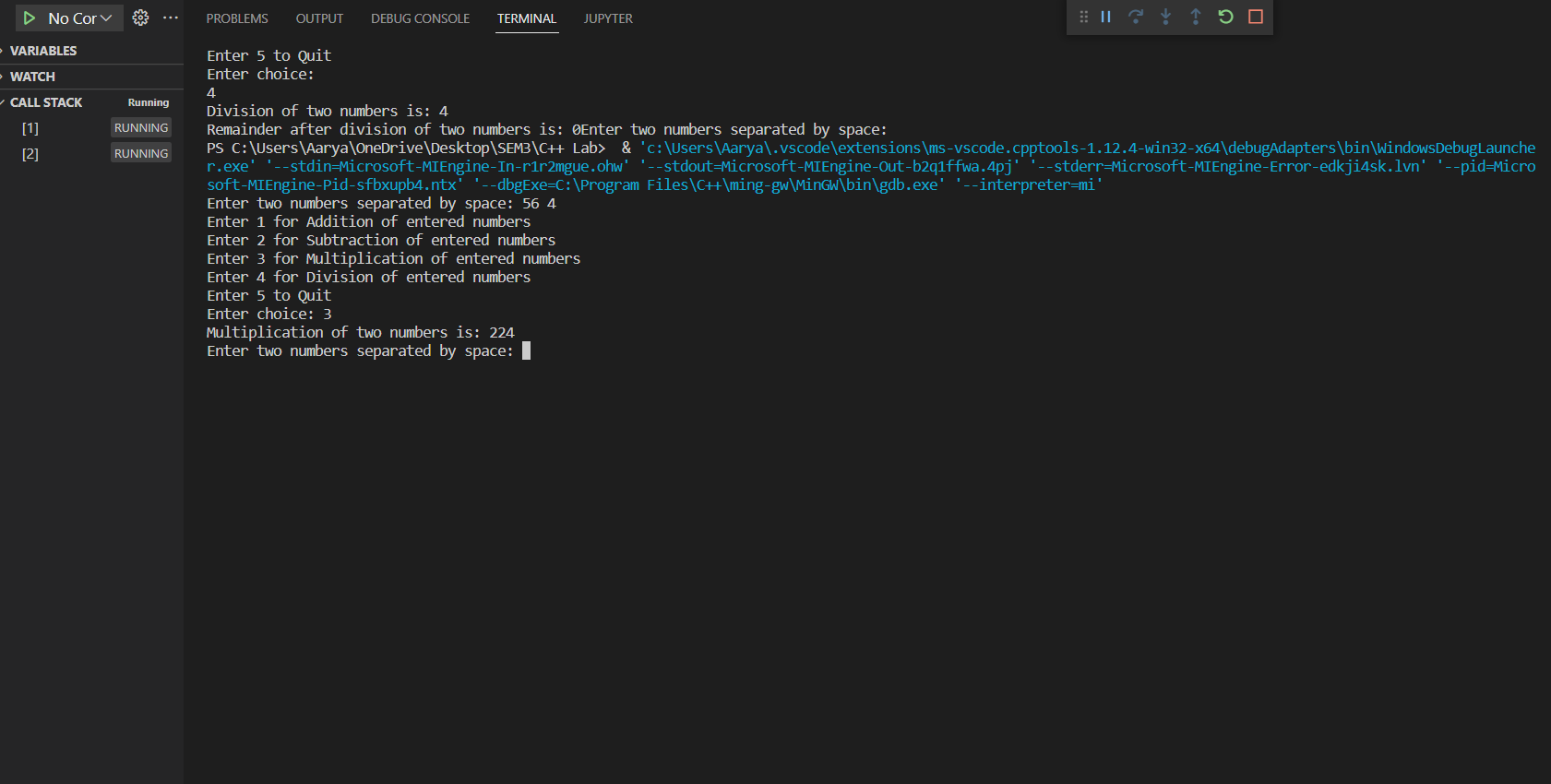
**}**

**}while(choice!=5);**

**return 0;**

**}**

**Output:**

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Test Cases (minimum 5 test cases required):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Sample Input** | **Sample Output** | **Description** | **Test Case Type (general/special)** | **Pass/Fail** |
| **1** | **10 , 5** | **2** | **Division** | **General** | **Pass** |
| **2** | **24 , 6** | **4** | **Division** | **General** | **Pass** |
| **3** | **56 , 4** | **224** | **Multiplication** | **General** | **Pass** |
| **4** | **23 , 4** | **19** | **Subtraction** | **General** | **Pass** |
| **5** | **3 , 2** | **5** | **Addition** | **General** | **Pass** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Questions:**

1. Compare the class in C++ with a structure in C

**Similarities:**

Both are container types, meaning that they contain other types as members.

Both have members, which can include constructors, methods, properties, fields, constants, enumerations, events, and event handlers.

Members of both can have individualized access levels. For example, one member can be declared Public and another Private.

Both can implement interfaces.

Both can have shared constructors, with or without parameters.

Both can expose a default property, provided that property takes at least one parameter.

Both can declare and raise events, and both can declare delegates.

**Differences:**

Structures are value types; classes are reference types. A variable of a structure type contains the structure's data, rather than containing a reference to the data as a class type does.

Structures use stack allocation; classes use heap allocation.

All structure elements are Public by default; class variables and constants are Private by default, while other class members are Public by default. This behavior for class members provides compatibility with the Visual Basic 6.0 system of defaults.

A structure must have at least one nonshared variable or nonshared, noncustom event element; a class can be completely empty.

Structure elements cannot be declared as Protected; class members can.

A structure procedure can handle events only if it is a SharedSub procedure, and only by means of the AddHandler Statement; any class procedure can handle events, using either the Handles keyword or the AddHandler statement. For more information.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Outcomes:**

**CO: 1.1 –**

C Vs. C++, C++ Basics: I/O in C++, Object-Oriented Thinking: Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts Abstraction, Encapsulation, Inheritance and Polymorphism.

**Conclusion: (Conclusion to be based on the outcomes achieved)**

We have learnt to implement a menu driven program based on the basics of the C++ Programming.

**Grade: AA / AB / BB / BC / CC / CD /DD**

Signature of faculty in-charge with date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**References:**

**Books/ Journals/ Websites:**

1. E Balagurusamy, Object oriented Programming with C++, Tata McGraw-Hill, 8th Edition September 2020
2. Herbert Schildt, C++: The Complete Reference, McGraw Hill Education, 4th edition, July 2017
3. Jeff Langr, Modern C++ Programming with Test-Driven Development : Code Better,Sleep Better, O′Reilly, 1st edition, November 2013
4. <https://docs.microsoft.com/en-us/cpp/cpp/?view=msvc-170>