Sahibkamaljot Singh

2061821

Contents

[REPORT 2](#_Toc49251542)

[Explanation 2](#_Toc49251543)

[Function overloading 2](#_Toc49251544)

[Inheritance 2](#_Toc49251545)

[Switch statements 3](#_Toc49251546)

[CLASS DIAGRAM 4](#_Toc49251547)

[CODE 4](#_Toc49251548)

[RESULTS 6](#_Toc49251549)

## REPORT

The project incorporates following concepts of C++ :

* Function overloading
* Inheritance
* Switch statements

### Explanation

Objective of the task is to take three inputs from the user, two of numeric datatype (i.e. float, double) and one character specifying the operation to be performed. These three variable are entered by the user and are two are passed to base class (Arithmetic) through each sub class or derived class. Values of these variables are assigned to protected variables of Arithmetic class or base class. Firstly, objects of derived classes are declared and these values are passed through these objects. Objective of derived class is to calculate the arithmetic operation. Derived classed do so by implementing an overload method of calculate (). Calculate method calculates the required operation and returns the value. Next, the switch operator comes into action and examines the operator command entered by the user. Switch operator calls the overloaded method of specific class through the object of that class and this method returns the value of operation into result variable.

### Function overloading

C++ provides with the facility of function overloading or method overloading. Function overloading means to declare different functions or methods with the same name but different attributes. In other words, function names could be same but function declaration should be different. In this example there are four functions with the same name of Calculate () but their definition is different from one another. In the following example, each function could be uniquely called through its class.

### Inheritance

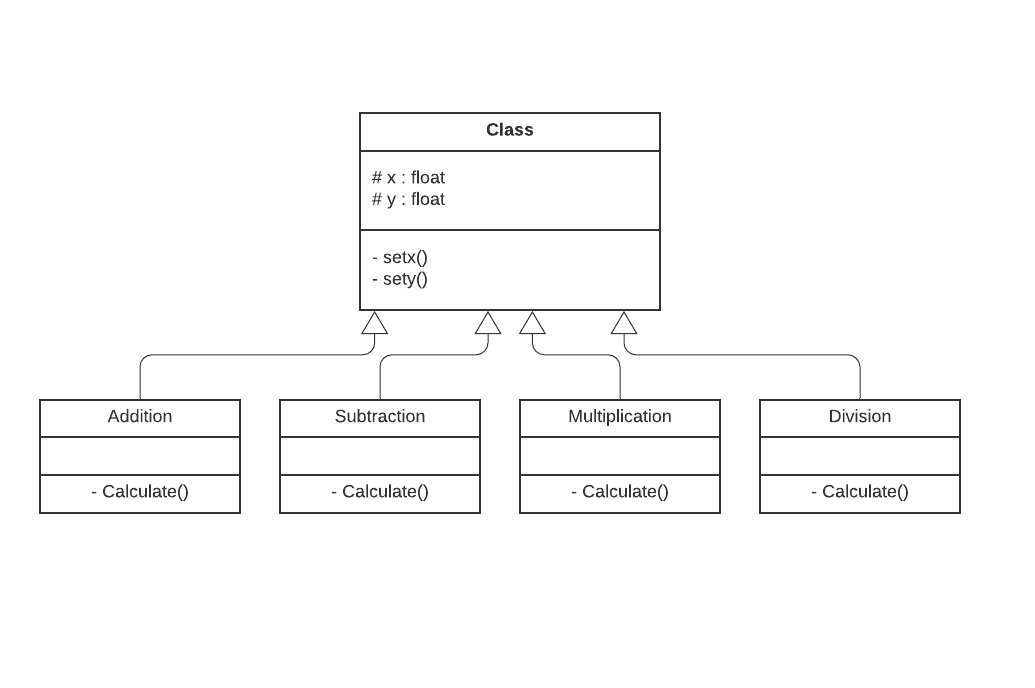
Inheritance plays a vital and core role in object oriented programming (OOP). There are two types of classes used in inheritance base class or super class and derived class or subclass. Sub classes inherit attributes or features of super class. Inheritance provides code usability and optimization. Problem of code duplication could be solved using inheritance.

Let us take an example of cell phones there are many types of cell phones in the market such as android phones, iOS phones, ambient phones and several others. If we create a base class of phone consisting all of the common or base features such as call(), text(), battery time(), display() and derive sub classes of android phones, iOS phones and ambient phones. All of these devices share the same base features from base class. Features of base class could be used by any sub class. Thus, cutting down code duplication.

### Switch statements

Long if else logics are difficult to implement. Switch statements provide an alternative to long if else statements. Switch statement controls the execution of the program based upon the underlying conditions. There are blocks of codes in switch statement referred as cases. Each case is checked against the condition and is executed or blocked according to the condition.

## CLASS DIAGRAM



## CODE

#include <iostream>

using namespace std;

class Arithmetic{ //BASE CLASS,,SUPER CLASS

public:

void setx(float t1)

{

x = t1;

}

void sety(float t2)

{

y = t2;

}

protected:

float x;

float y;

};

class Addition : public Arithmetic{ // SUBCLASS,,DERIVED CLASS

public:

float calculate() // METHOD OVERLOADING

{

return (x + y);

}

};

class Subtraction : public Arithmetic{ // SUBCLASS,,DERIVED CLASS

public:

float calculate() // METHOD OVERLOADING

{

return (x - y);

}

};

class Multiplication : public Arithmetic{ // SUBCLASS,,DERIVED CLASS

public:

float calculate() // METHOD OVERLOADING

{

return (x \* y);

}

};

class Division : public Arithmetic{ // SUBCLASS,,DERIVED CLASS

public:

float calculate() // METHOD OVERLOADING

{

return (x / y);

}

};

int main()

{

float number1;

float number2;

float result;

char operatoor;

cout<<"ENTER FIRST NUMBER \n";

cin>>number1;

cout<<"\nENTER SECOND NUMBER\n";

cin>>number2;

cout<<"\nENTER OPERATOR\n";

cin>>operatoor;

Addition a;

a.setx(number1);

a.sety(number2);

Subtraction s;

s.setx(number1);

s.sety(number2);

Multiplication m;

m.setx(number1);

m.sety(number2);

Division d;

d.setx(number1);

d.sety(number2);

switch(operatoor){ //SWITCH CASE

case '+' :

result = a.calculate();

break;

case '-':

result = s.calculate();

break;

case '\*':

result = m.calculate();

break;

case '/':

result = d.calculate();

break;

}

cout<<"\nRESULT OF OPERATION : "<<result;

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

return 0;

}

## RESULTS

