Jaypee University of Engineering and Technology, Guna

Department of Computer Science and Engineering

Object Oriented Programming Lab
(18B17CI271)

Lab Exercise-5

Constructors and Destructors

Name : Palash Mishra

Enroll no. : 201B172

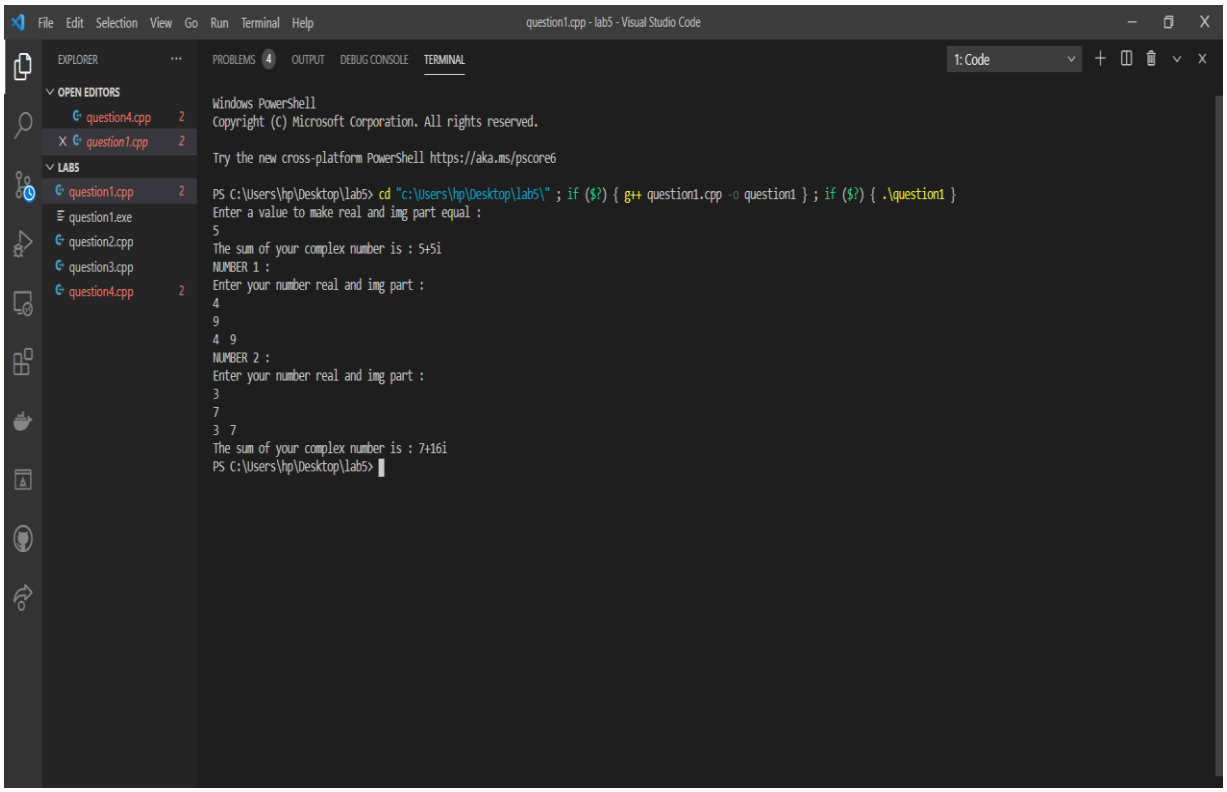
QUESTION 1 :

Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and imaginary parts to equal values and third which takes two argument is used to initialized real and imaginary to two different values. Include sum member function that takes two objects and performs the addition of these two objects. Write display function to display the object in complex form i.e. 2+i6 . Write main function to test your program.

```
#include <iostream>
using namespace std;
class complex
{
public:
    int x, y;
    complex()
    {
        x = 0;
        y = 0;
    }
    complex(int t)
    {
        x = t;
        y = t;

        cout << "The sum of your complex number is : " << x << "+" << y << "i"
 << endl;
    }
    complex(int r, int i)
    {
        x = r;
        y = i;
        cout << x << "  " << y << endl;
    }

    void sum(complex &ob1, complex &ob2)
    {

        x = ob1.x + ob2.x;
        y = ob1.y + ob2.y;
    }
    void display()
    {

        cout << "The sum of your complex number is : " << x << "+" << y << "i"
 << endl;
    }
```

```
};
int main()
{
    int real, img, value;
    complex fi;
    cout << "Enter a value to make real and img part equal :\n";
    cin >> value;
    complex sec(value);
    cout << "NUMBER 1 : \n";
    cout << "Enter your number real and img part :\n";
    cin >> real >> img;
    complex thi_1(real, img);
    cout << "NUMBER 2 : \n";
    cout << "Enter your number real and img part :\n";
    cin >> real >> img;
    complex thi_2(real, img);
    thi_1.sum(thi_1, thi_2);
    thi_1.display();

    return 0;
}
```

OUTPUT :

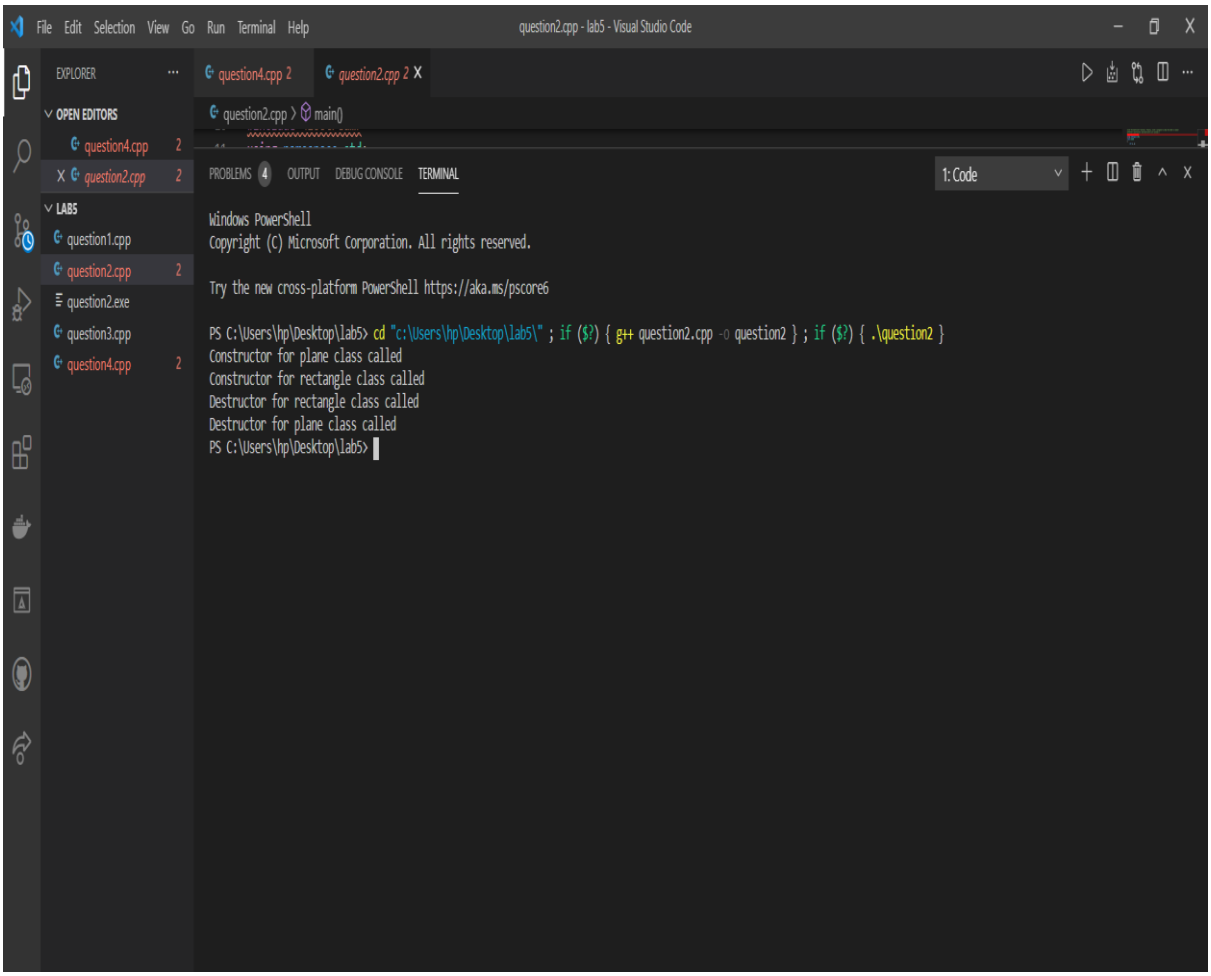


QUESTION 2 :

A point in a two-dimensional plane having coordinate as (x,y), can be represented by a class whose private data members are x and y. Write the constructor and destructor functions of the class. The constructor should initialise (x,y) by passing parameters values. Now, a rectangle can be represented by the top-left and bottom-right vertices. Define a class say 'Rectangle' whose private data members are top-left and bottom-right vertices. Write the parameterised constructor function of the class 'Rectangle'. Also, write the destructor function. Finally, write a program to show the order in which different constructors and destructors are called.

```
#include <iostream>
using namespace std;
class plane
{
    int x, y;

public:
    plane(int x, int y)
    {
        cout << "Constructor for plane class called \n";
    }
    ~plane()
    {
        cout << "Destructor for plane class called \n";
    }
};
class rectangle
{
    int top_l, bottom_r;

public:
    rectangle(int a, int b)
    {
        cout << "Constructor for rectangle class called \n";
    }
    ~rectangle()
    {
        cout << "Destructor for rectangle class called \n";
    }
};

int main()
{
    class plane pl(2, 3);
    class rectangle rc(5,9);
}
```

OUTPUT :

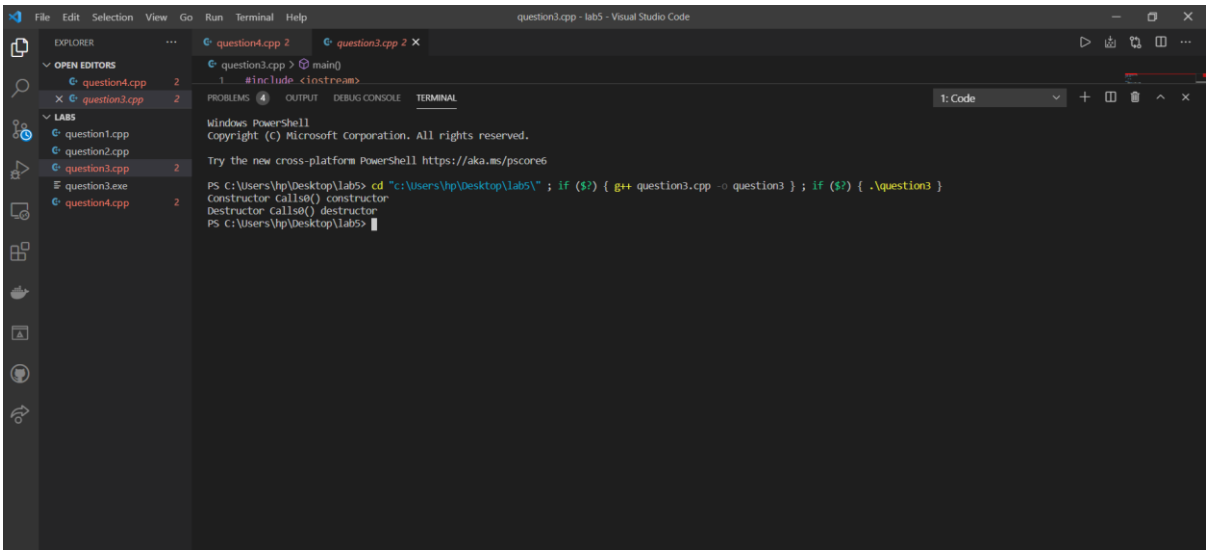

File Edit Selection View Go Run Terminal Help
question2.cpp - lab5 - Visual Studio Code
EXPLORER
question4.cpp 2
question2.cpp 2
OPEN EDITORS
question4.cpp 2
question2.cpp 2
LABS
question1.cpp
question2.cpp 2
question2.exe
question3.cpp
question4.cpp 2
question2.cpp > main()
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
1: Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\hp\Desktop\lab5> cd "c:\Users\hp\Desktop\lab5\" ; if ($?) { g++ question2.cpp -o question2 } ; if ($?) { .\question2 }
Constructor for plane class called
Constructor for rectangle class called
Destructor for rectangle class called
Destructor for plane class called
PS C:\Users\hp\Desktop\lab5>

QUESTION 3 :

Write a program to show that, the constructor and destructor functions of a globally declared object are the first and last functions, respectively to be called in a program.

```
#include <iostream>
using namespace std;
class oop
{
    int x;

public:
    oop(int y = 0) : x(y)
    {
        cout << "Constructor Calls" << x << "() constructor " << endl;
    }
    ~oop()
    {
        cout << "Destructor Calls" << x << "() destructor " << endl;
    }
};
int main()
{
    oop m;
}
```

OUTPUT :

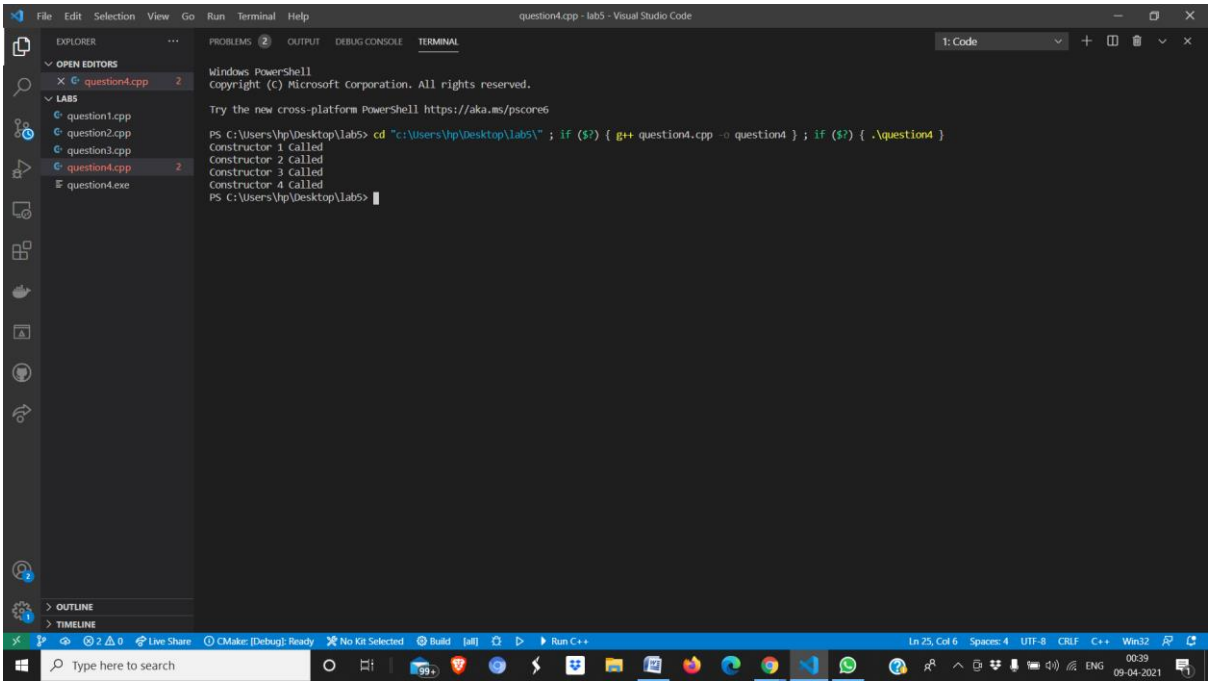


QUESTION 4 :

Write a program to show that constructors follow the property of function overloading as well as default parameter. Also, show that in case of constructor also, default parameter may create problem in implementing function overloading.

```
/*Write a program to show that constructors follow the property of function ov
erloading as
well as default parameter. Also, show that in case of constructor also, defaul
t parameter
may create problem in implementing function overloading.*/
#include <iostream>
using namespace std;
class cons
{
    int x;

public:
    cons()
    {
        cout << "Constructor 1 Called" << endl;
    }
    cons(int y)
    {
        cout << "Constructor 2 Called" << endl;
    }
    cons(char c)
    {

        cout << "Constructor 3 Called" << endl;
    }
    cons(double y)
    {
        cout << "Constructor 4 Called" << endl;
    }
    cons(int y=0)
    {
        cout << "Constructor 1 Called" << endl;
    }
};
int main()
{
    cons m;
    cons n(4);
    cons o('p');
    cons p(5.8);
    cons q(5);
}
```

OUTPUT :



Problem with using default parameter in implementation of function overloading