Object Oriented Programming

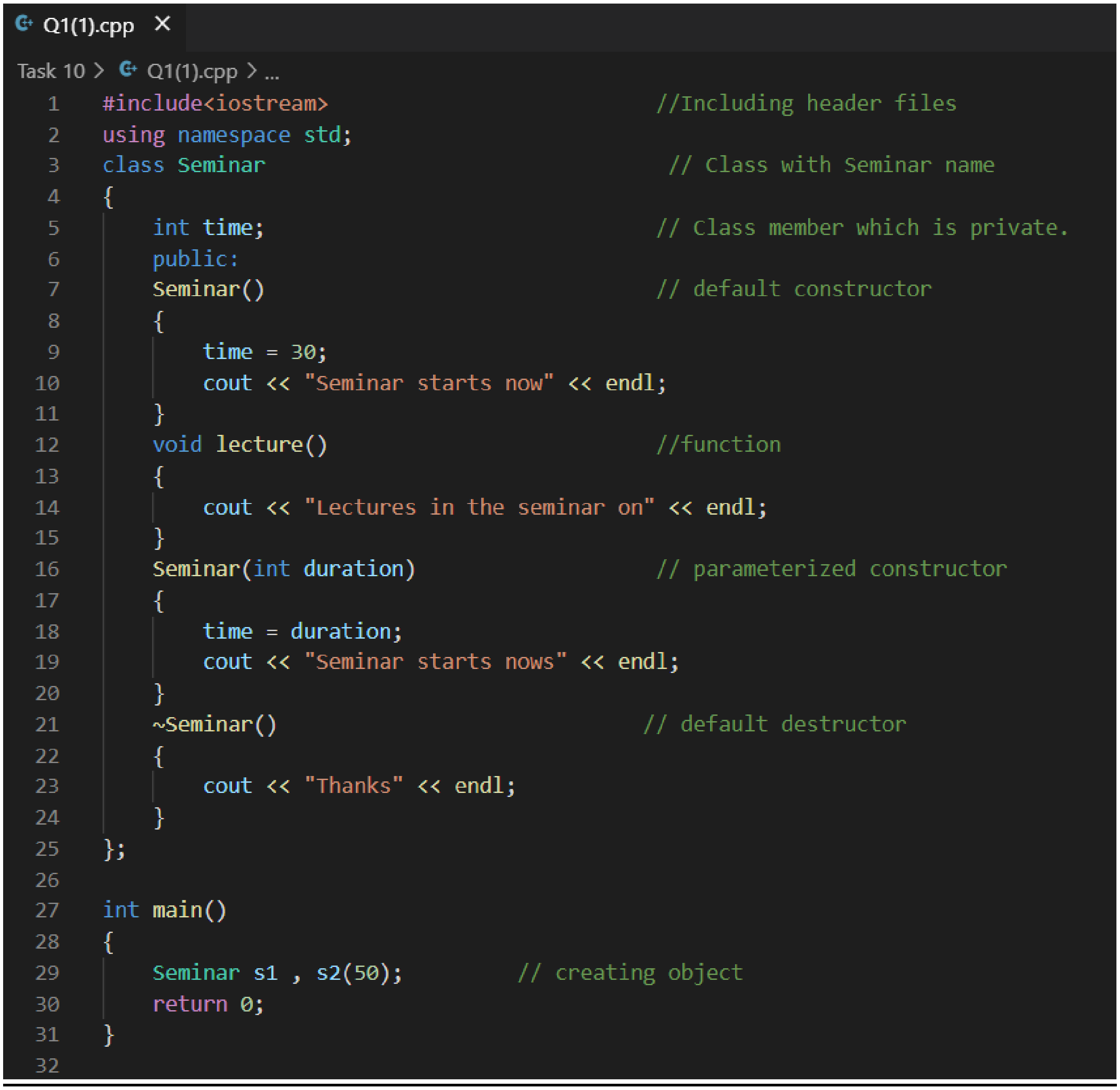
Lab task #10

Submitted by
Saad Ahmad
(20P-0051)

Submitted to
Mr. Muhammad Abdullah
(Computer Instructor)

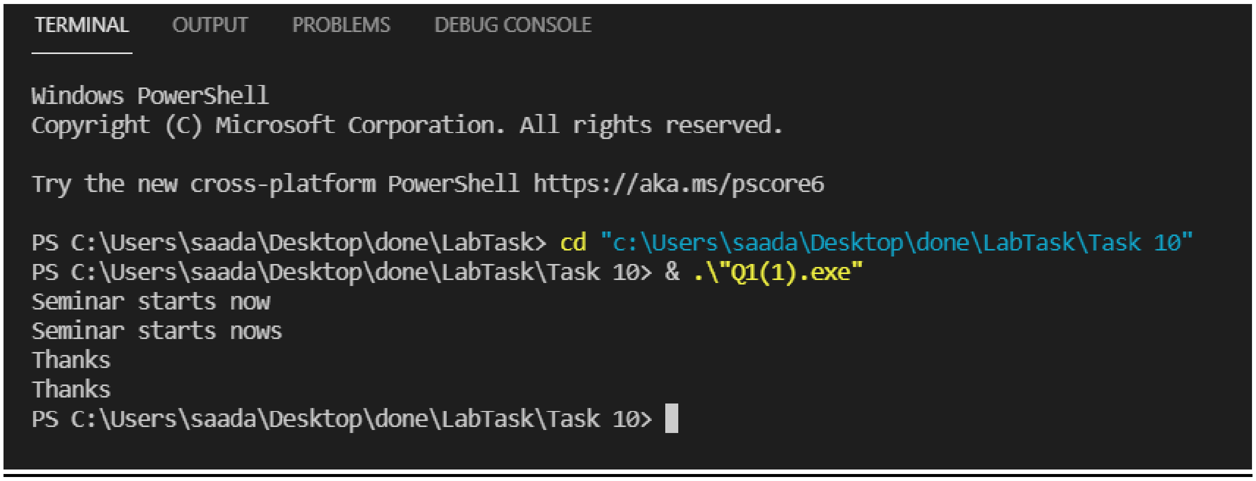
Question #1

i) **Code:**



```cpp
#include<iostream>                          //Including header files
using namespace std;
class Seminar                                // Class with Seminar name
{
    int time;                               // Class member which is private.
    public:
    Seminar()                               // default constructor
    {
        time = 30;
        cout << "Seminar starts now" << endl;
    }
    void lecture()                          //function
    {
        cout << "Lectures in the seminar on" << endl;
    }
    Seminar(int duration)                   // parameterized constructor
    {
        time = duration;
        cout << "Seminar starts nows" << endl;
    }
    ~Seminar()                             // default destructor
    {
        cout << "Thanks" << endl;
    }
};

int main()
{
    Seminar s1 , s2(50);         // creating object
    return 0;
}

```

Output:



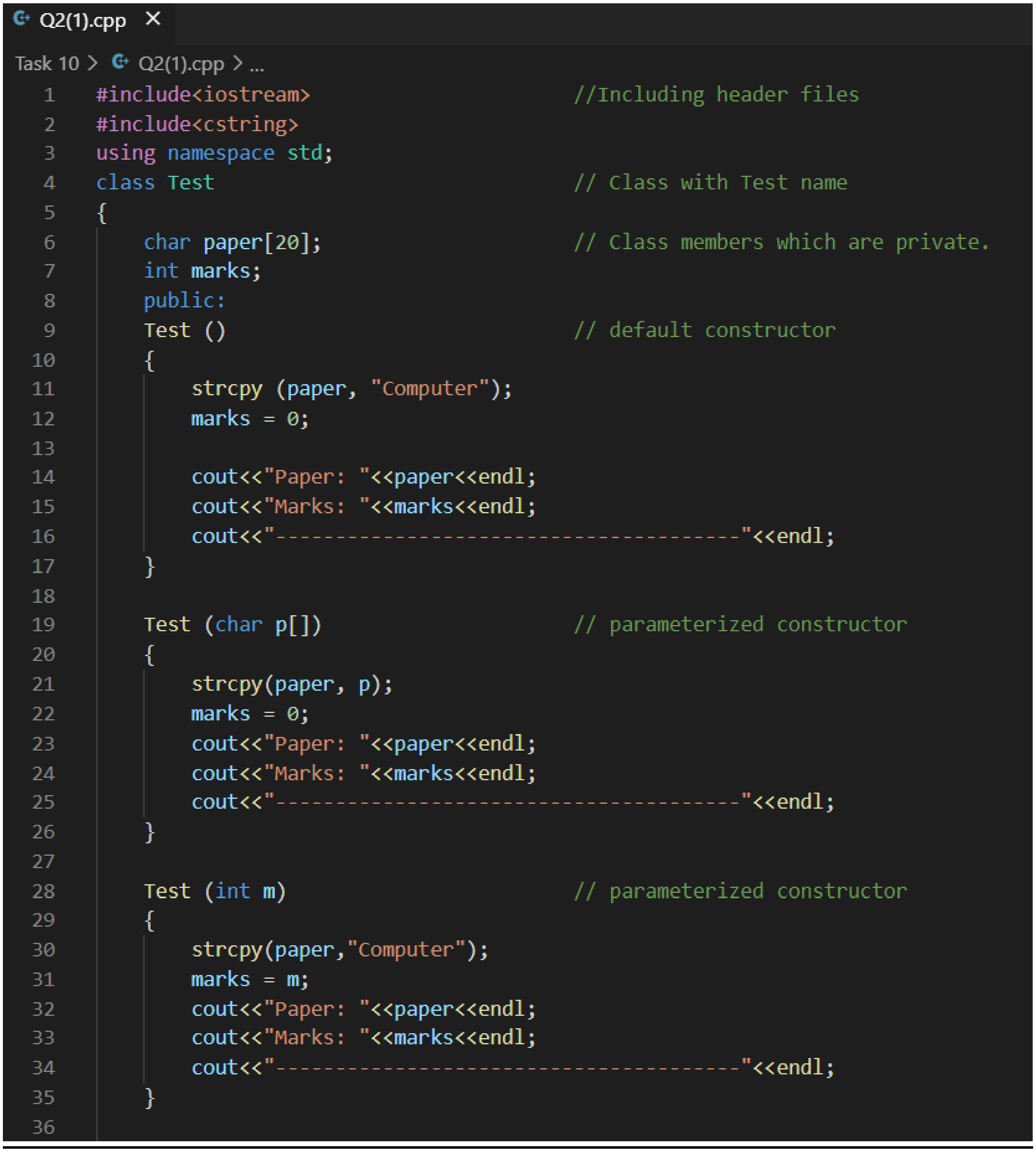
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\saada\Desktop\done\LabTask> cd "c:\Users\saada\Desktop\done\LabTask\Task 10"
PS C:\Users\saada\Desktop\done\LabTask\Task 10> & .\"Q1(1).exe"
Seminar starts now
Seminar starts nows
Thanks
Thanks
PS C:\Users\saada\Desktop\done\LabTask\Task 10>
```

ii) Function 4 is a destructor, which is invoked when an object is going to be killed.

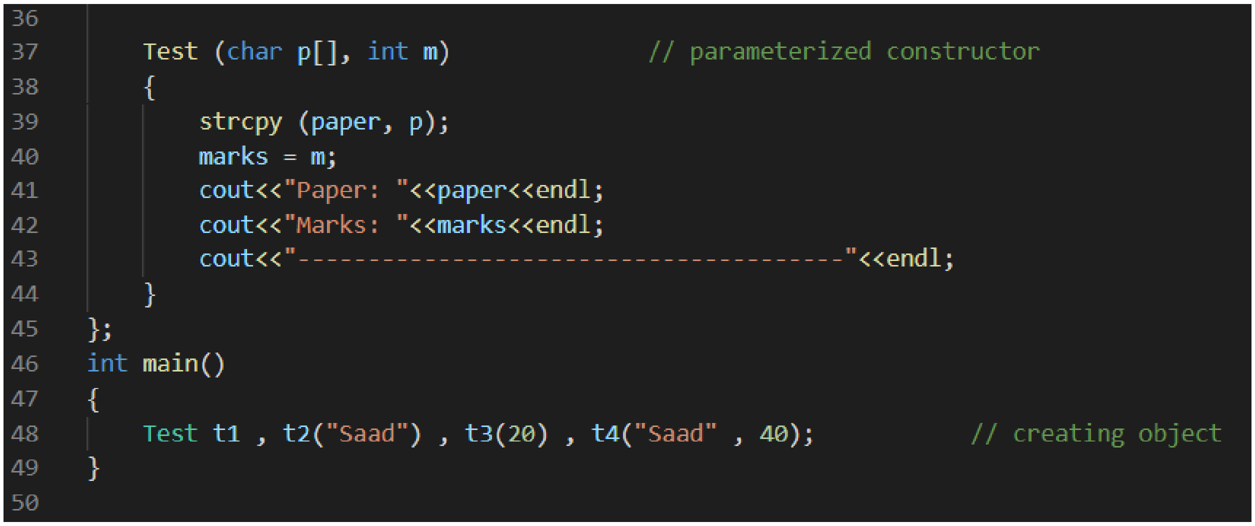
iii) Function 1 and 3 are constructors, function 1 is a default constructor and function 3 is parametrized constructor. When we make an object without any value then default constructor will be called/invoked and when we make an object with a passing value then parameterized constructor will be called/invoked.

Question #2

i) **Code:**

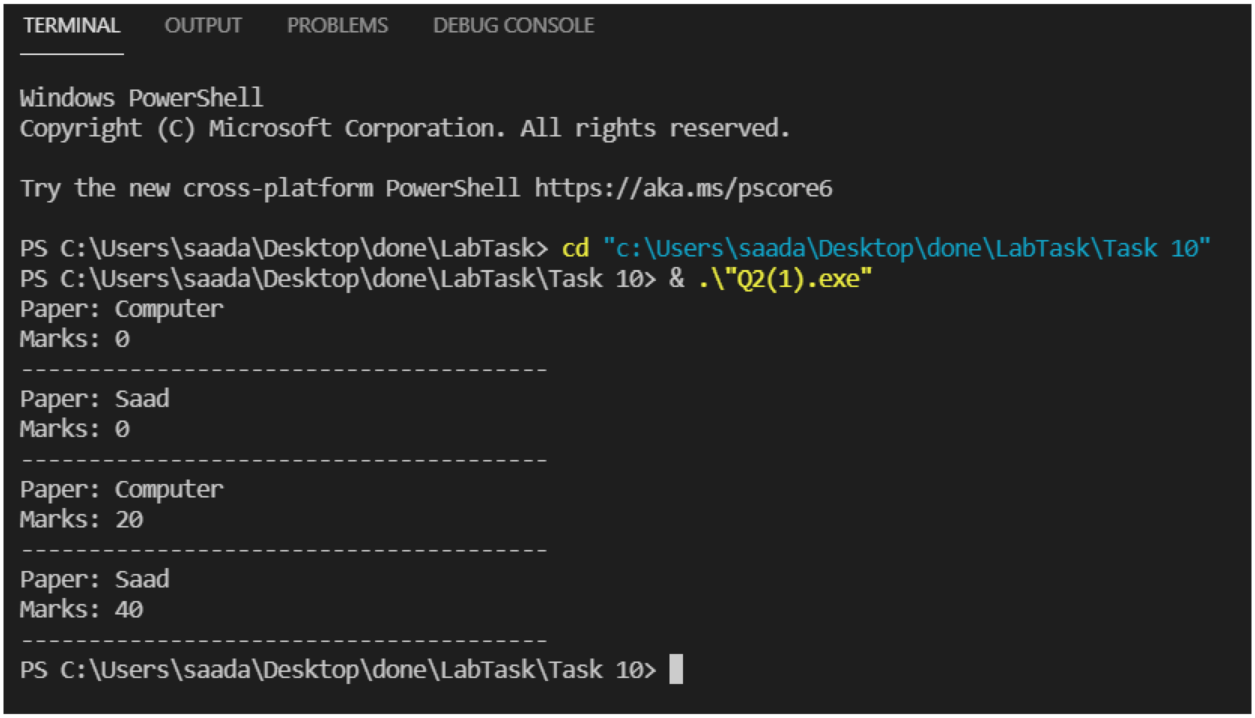


```cpp
#include<iostream>                      //Including header files
#include<cstring>
using namespace std;
class Test                              // Class with Test name
{
    char paper[20];                     // Class members which are private.
    int marks;
    public:
    Test ()                             // default constructor
    {
        strcpy (paper, "Computer");
        marks = 0;

        cout<<"Paper: "<<paper<<endl;
        cout<<"Marks: "<<marks<<endl;
        cout<<"---------------------------------------"<<endl;
    }

    Test (char p[])                     // parameterized constructor
    {
        strcpy(paper, p);
        marks = 0;
        cout<<"Paper: "<<paper<<endl;
        cout<<"Marks: "<<marks<<endl;
        cout<<"---------------------------------------"<<endl;
    }

    Test (int m)                        // parameterized constructor
    {
        strcpy(paper,"Computer");
        marks = m;
        cout<<"Paper: "<<paper<<endl;
        cout<<"Marks: "<<marks<<endl;
        cout<<"---------------------------------------"<<endl;
    }

```



```cpp

    Test (char p[], int m)              // parameterized constructor
    {
        strcpy (paper, p);
        marks = m;
        cout<<"Paper: "<<paper<<endl;
        cout<<"Marks: "<<marks<<endl;
        cout<<"----------------------------------------"<<endl;
    }
};
int main()
{
    Test t1 , t2("Saad") , t3(20) , t4("Saad" , 40);           // creating object
}

```

Output:

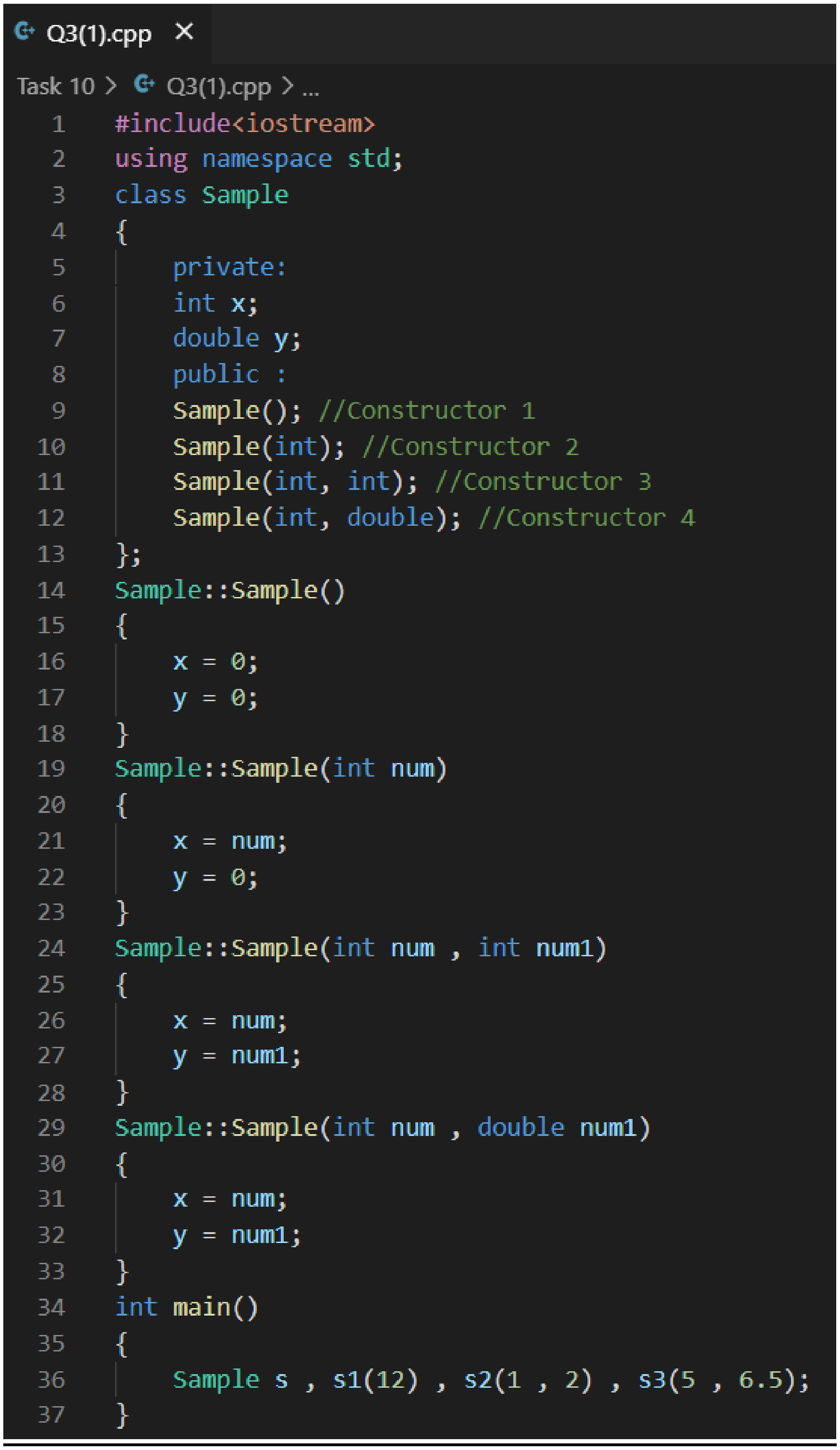


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\saada\Desktop\done\LabTask> cd "c:\Users\saada\Desktop\done\LabTask\Task 10"
PS C:\Users\saada\Desktop\done\LabTask\Task 10> & .\"Q2(1).exe"
Paper: Computer
Marks: 0
----------------------------------------
Paper: Saad
Marks: 0
----------------------------------------
Paper: Computer
Marks: 20
----------------------------------------
Paper: Saad
Marks: 40
----------------------------------------
PS C:\Users\saada\Desktop\done\LabTask\Task 10>
```

ii) Feature of Object Oriented Programming, which is demonstrated using Function 1, Function 2, Function 3 and Function 4 is constructor overloading. When multiple constructors are used in the same class then it is called Constructor Overloading. It gives us multiple ways to initialize objects in a class. It increases flexibility by having multiple constructors in a single class. A constructor is called depending upon the number and type of arguments passed.

Question #3

Code:



```cpp
#include<iostream>
using namespace std;
class Sample
{
    private:
    int x;
    double y;
    public :
    Sample(); //Constructor 1
    Sample(int); //Constructor 2
    Sample(int, int); //Constructor 3
    Sample(int, double); //Constructor 4
};
Sample::Sample()
{
    x = 0;
    y = 0;
}
Sample::Sample(int num)
{
    x = num;
    y = 0;
}
Sample::Sample(int num , int num1)
{
    x = num;
    y = num1;
}
Sample::Sample(int num , double num1)
{
    x = num;
    y = num1;
}
int main()
{
    Sample s , s1(12) , s2(1 , 2) , s3(5 , 6.5);
}
```