Lab Assingment-4

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 19/08/21

QUES 1: Create a class which stores name, roll number and total marks for a student. Input data for n students. Find the average marks scored by n students, store it as a data member of the class and display it using a function which may be called without object.

SOLUTION:

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

class Student
{
public:
    char name_535[20];
    int roll_535;
    float totalmarks_535;
    float average_535;
};

void insert(Student &student)
{
    cout << "Enter name: ";
    cin >> student.name_535;
    cout << "Enter Roll: ";
    cin >> student.roll_535;
    cout << "Enter total marks: ";
    cin >> student.totalmarks_535;
}

void average(Student student[], int n)
{
    float sum_535 = 0;
    for (int i = 0; i < n; i++)
    {
        sum_535 += student[i].totalmarks_535;
    }

    for (int i = 0; i < n; i++)
    {
        student[i].average_535 = sum_535 / n;
    }
}

void display(Student student)
{
    cout << "Name: " << student.name_535 << endl;
    cout << "Roll: " << student.roll_535 << endl;
    cout << "Total Marks: " << student.totalmarks_535 << endl;
    cout << "Average: " << student.average_535 << endl;
}

int main()
{
    Student student_535[10];
    int n;
    cout << "Enter number of students: ";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cout << "Student no.: " << i + 1 << endl;
        insert(student_535[i]);
    }
    average(student_535, n);
    for (int i = 0; i < n; i++)
    {
        cout << "Student no.: " << i + 1 << endl;
        display(student_535[i]);
        cout << endl;
    }
    return 0;
}
```

OUTPUT:

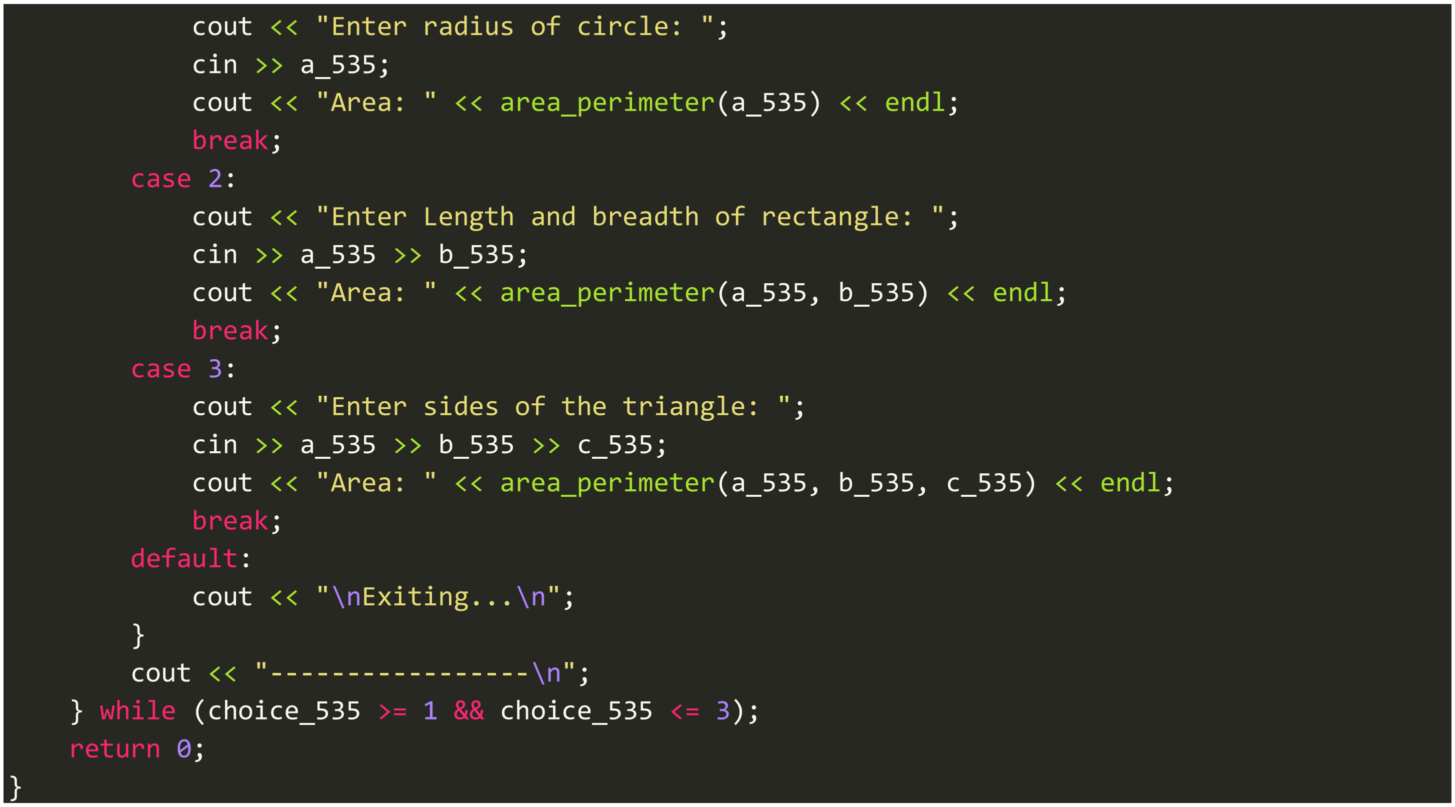
```
Enter number of students: 3
Student no.: 1
Enter name: Windows
Enter Roll: 11
Enter total marks: 480
Student no.: 2
Enter name: Ubuntu
Enter Roll: 21
Enter total marks: 490
Student no.: 3
Enter name: MacOS
Enter Roll: 15
Enter total marks: 470
Student no.: 1
Name: Windows
Roll: 11
Total Marks: 480
Average: 480

Student no.: 2
Name: Ubuntu
Roll: 21
Total Marks: 490
```

```
Average: 480

Student no.: 3
Name: MacOS
Roll: 15
Total Marks: 470
Average: 480
```

QUES 2: WAP to find area of a circle, a rectangle and a triangle, using concept of function
overloading.

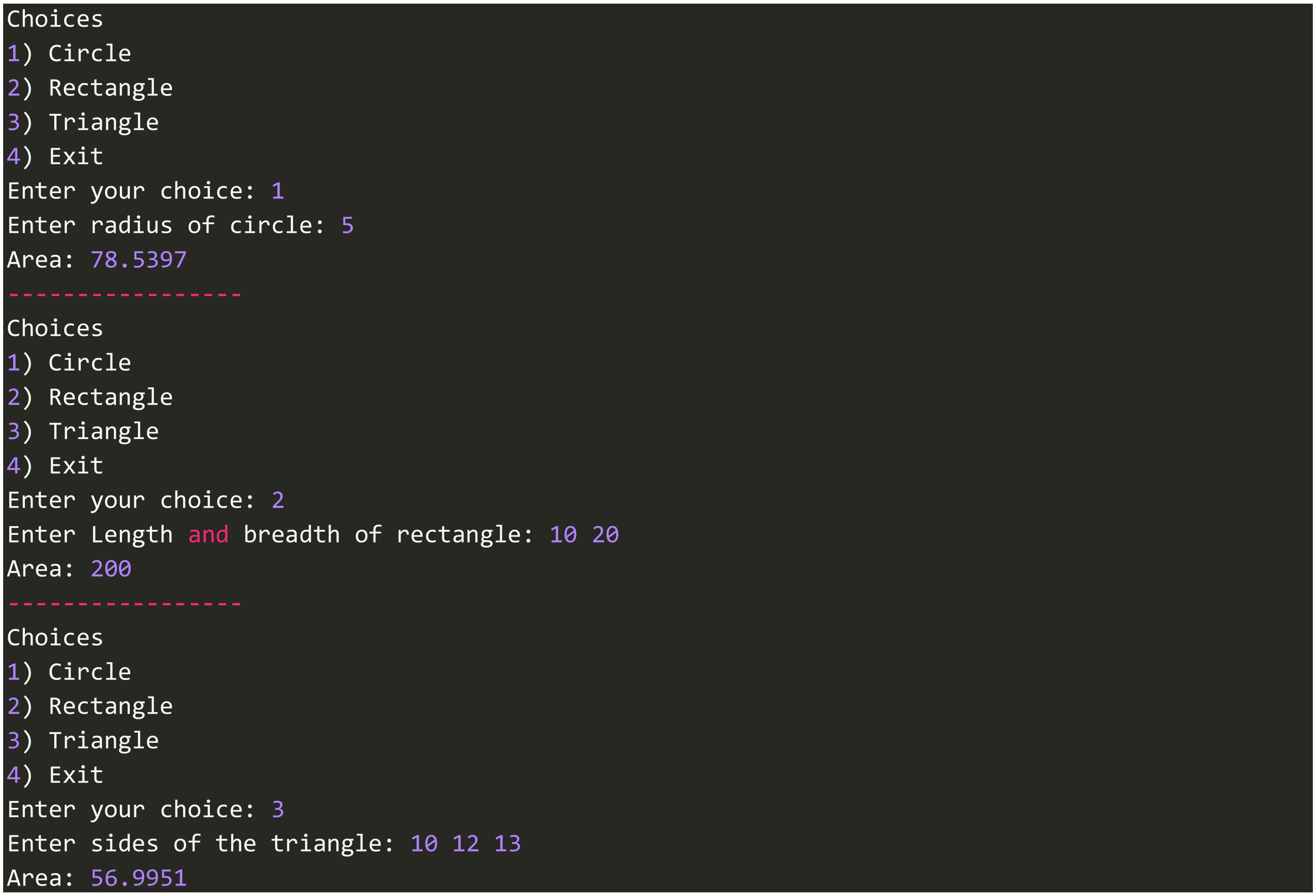
SOLUTION:

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

float area_perimeter(float r)
{

    float area_535 = 3.14159 * r * r;
    return area_535;
}

float area_perimeter(float l, float w)
{
    float area_535 = l * w;
    return area_535;
}

float area_perimeter(float a, float b, float c)
{
    float s_535 = (a + b + c) / 2;
    float area_535 = sqrt(s_535 * (s_535 - a) * (s_535 - b) * (s_535 - c));
    return area_535;
}

int main()
{
    float a_535, b_535, c_535;
    int choice_535;
    do
    {
        cout << "Choices\n1) Circle\n";
        cout << "2) Rectangle\n3) Triangle\n4) Exit\nEnter your choice: ";
        cin >> choice_535;
        switch (choice_535)
        {
        case 1:
```



```
            cout << "Enter radius of circle: ";
            cin >> a_535;
            cout << "Area: " << area_perimeter(a_535) << endl;
            break;
        case 2:
            cout << "Enter Length and breadth of rectangle: ";
            cin >> a_535 >> b_535;
            cout << "Area: " << area_perimeter(a_535, b_535) << endl;
            break;
        case 3:
            cout << "Enter sides of the triangle: ";
            cin >> a_535 >> b_535 >> c_535;
            cout << "Area: " << area_perimeter(a_535, b_535, c_535) << endl;
            break;
        default:
            cout << "\nExiting...\n";
        }
        cout << "-----------------\n";
    } while (choice_535 >= 1 && choice_535 <= 3);
    return 0;
}
```

SOLUTION:



```
Choices
1) Circle
2) Rectangle
3) Triangle
4) Exit
Enter your choice: 1
Enter radius of circle: 5
Area: 78.5397
-----------------
Choices
1) Circle
2) Rectangle
3) Triangle
4) Exit
Enter your choice: 2
Enter Length and breadth of rectangle: 10 20
Area: 200
-----------------
Choices
1) Circle
2) Rectangle
3) Triangle
4) Exit
Enter your choice: 3
Enter sides of the triangle: 10 12 13
Area: 56.9951
```

```
-----------------
Choices
1) Circle
2) Rectangle
3) Triangle
4) Exit
Enter your choice: 4

Exiting...
-----------------
```

QUES 3: WAP to find square and cube of a number using inline function.

SOLUTION:

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

inline int square(int x)
{
    return (x * x);
}

inline int cube(int x)
{
    return (x * x * x);
}

int main()
{
    cout << "Square: " << square(4);
    cout << "\nCube: " << cube(5);
    return 0;
}
```

OUTPUT:

```
Square: 16
Cube: 125
```

QUES 4: WAP to swap two integers using pass by reference.

SOLUTION:

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

void ref(int &x, int &y)
{
    int temp = x;
    x = y;
    y = temp;
}

int main()
{
    int a, b;

    cout << "\nEnter two numbers to be swapped: ";
    cin >> a >> b;

    cout << "Swapping using call by reference.....\n";
    ref(a, b);
    cout << a << " " << b;
    return 0;
}
```

OUTPUT:

```
Enter two numbers to be swapped: 10 20
Swapping using call by reference.....
20 10
```

QUES 5: WAP to create a functions to calculate simple interest and compound interest by using function overloading concept.

SOLUTION:

```
#include <iostream>
#include <math.h>
using namespace std;

void interest(float pri, int time, float rate = 7.5)
{
    float si = (pri * rate * time) / 100;
    cout << "Simple Interest: " << si;
}

void interest(float pri, float time, float rate)
{
    float ci = pri * pow((1 + rate / 100), time);
    cout << "Compound Interest: " << ci;
}

int main()
{
    int choice;

    cout << "1. Simple Interest\n";
    cout << "2. Compound Interest\n";
    cout << "Enter your choice: ";
    cin >> choice;

    switch (choice)
    {
    case 1:
        int time;
        float pri, rate;
        cout << "Enter principle, rate of interest and time: ";
        cin >> pri >> rate >> time;
        interest(pri, time, rate);
        break;
    case 2:
        float p, r, t;
        cout << "Enter principle, rate of interest and time: ";
        cin >> p >> r >> t;
        interest(p, t, r);
        break;

    default:
        cout << "Wrong Choice!";
        break;
    }
    return 0;
}
```

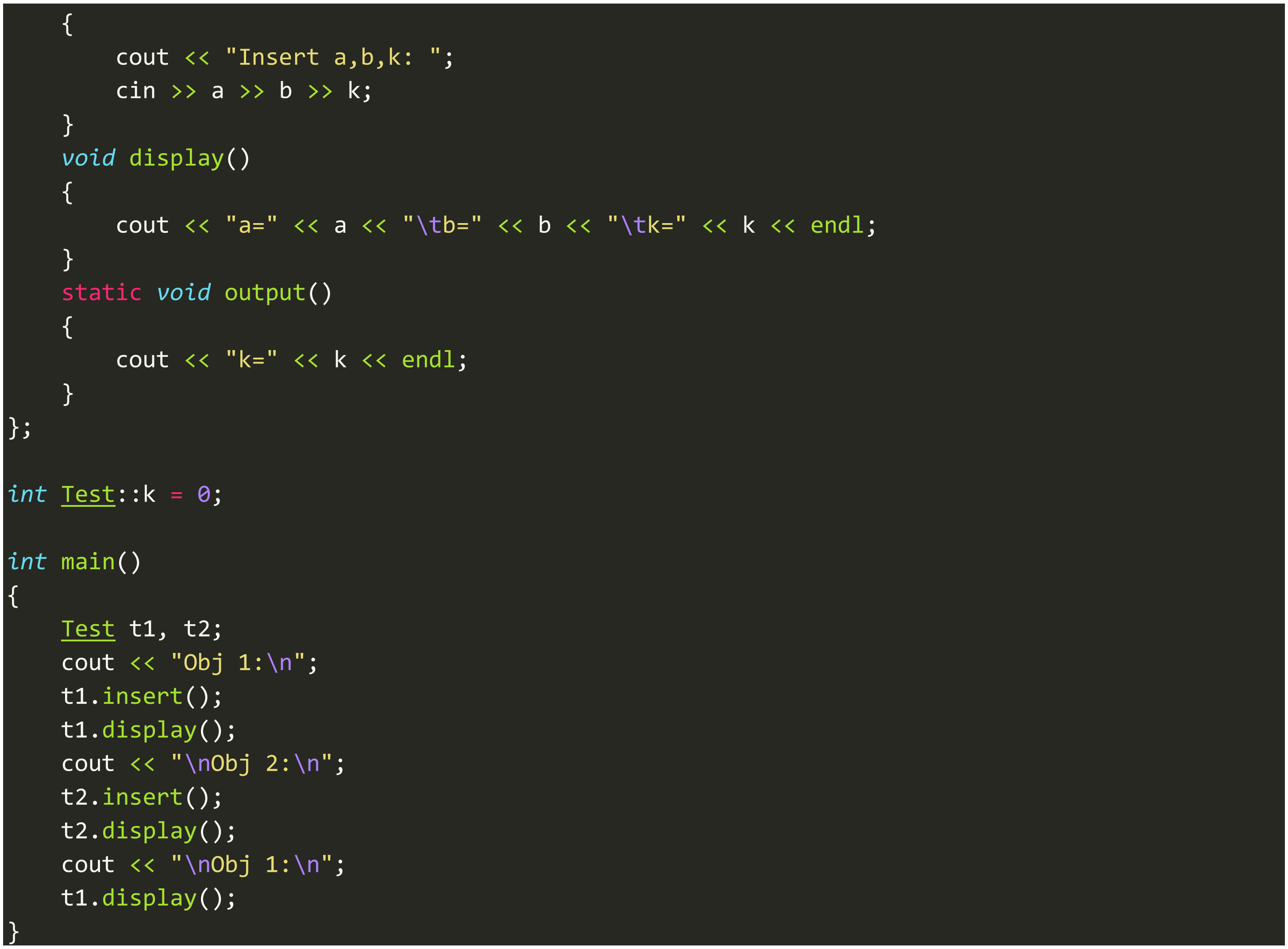
OUTPUT:

```
1. Simple Interest
2. Compound Interest
Enter your choice: 1
Enter principle, rate of interest and time: 120000 10 5
Simple Interest: 60000
```

QUES 6: Static data members of a class occupy memory once whereas non static data members occupy members as per the number of objects created. Justify the statement by writing a program.

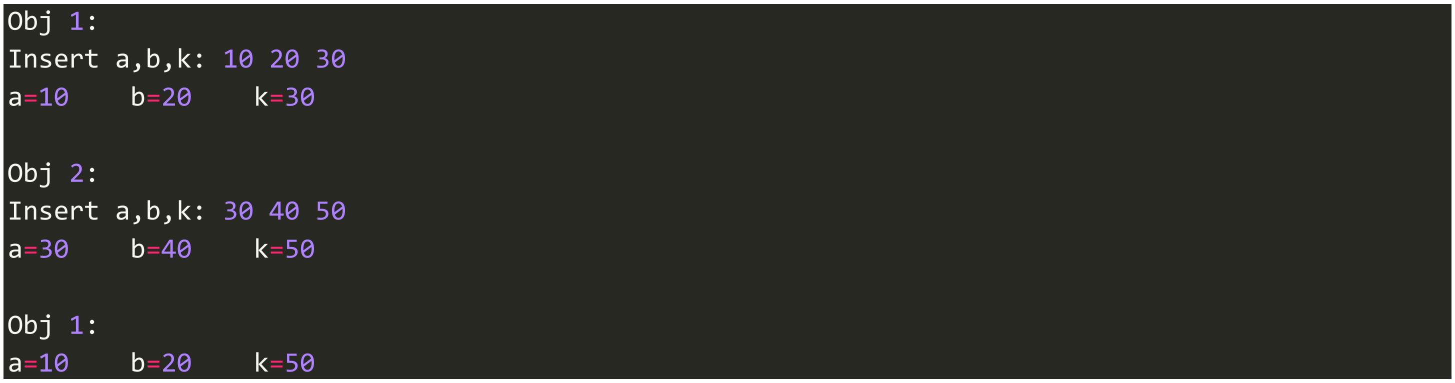
SOLUTION:

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

class Test
{
private:
    int a, b;
    static int k;

public:
    void insert()
```



```
    {
        cout << "Insert a,b,k: ";
        cin >> a >> b >> k;
    }
    void display()
    {
        cout << "a=" << a << "\tb=" << b << "\tk=" << k << endl;
    }
    static void output()
    {
        cout << "k=" << k << endl;
    }
};

int Test::k = 0;

int main()
{
    Test t1, t2;
    cout << "Obj 1:\n";
    t1.insert();
    t1.display();
    cout << "\nObj 2:\n";
    t2.insert();
    t2.display();
    cout << "\nObj 1:\n";
    t1.display();
}
```

OUTPUT:



```
Obj 1:
Insert a,b,k: 10 20 30
a=10    b=20    k=30

Obj 2:
Insert a,b,k: 30 40 50
a=30    b=40    k=50

Obj 1:
a=10    b=20    k=50
```