LAB ASSIGNMENT - 1

ARZOO KHAN

(20MCA011)

MCA

(Semester - II)

May 26 , 2021

—

CSC26: Lab – III (OOP)

—

Dr. S. Zeeshan Hussain

(Associate Professor, D/o Computer Science)

1. ***Write a program to implement the usage of static data members and static member functions of a class.***

SOURCE CODE:

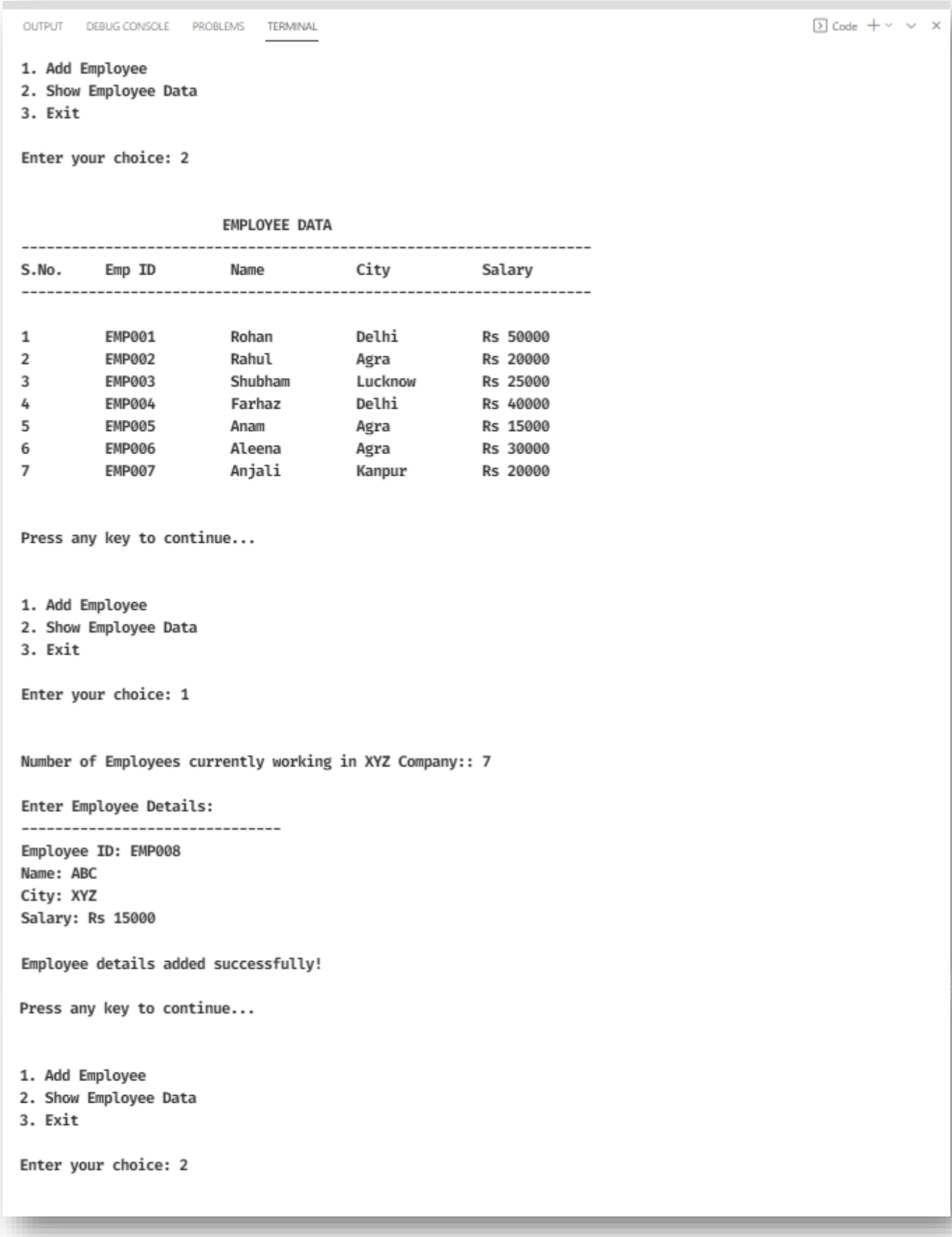
```
#include <iostream>
using namespace std;
#include <iomanip>
#include <string>
#include <conio.h>

class Employee
{
private:
    string id;
    string name;
    string city;
    float salary;
    static int NoOfEmployees;

public:
    static float get_NoOfEmplyees(void);
    friend void showEmployeeData(Employee *);
    void setData(string, string, string, float);
    void getData(void);
    void setSalary(float);
    float increaseSalary(float);
};


// Static data member of Employee class
int Employee::NoOfEmployees = 0;


// Static member function to get No of Employees
float Employee ::get_NoOfEmplyees(void)
{
    return NoOfEmployees;
}


//Driver Code
int main(void)
{
    Employee emp[10];
```

```
int empCount, choice;

emp[0].setData("EMP001", "Rohan", "Delhi", 50000);
emp[1].setData("EMP002", "Rahul", "Agra", 20000);
emp[2].setData("EMP003", "Shubham", "Lucknow", 25000);
emp[3].setData("EMP004", "Farhaz", "Delhi", 40000);
emp[4].setData("EMP005", "Anam", "Agra", 15000);
emp[5].setData("EMP006", "Aleena", "Agra", 30000);
emp[6].setData("EMP007", "Anjali", "Kanpur", 20000);

while (true)
{
    cout << "\n\n\n1. Add Employee"
         << "\n2. Show Employee Data"
         << "\n3. Exit"
         << "\n\nEnter your choice: ";
    cin >> choice;

    switch (choice)
    {
    case 1:

        // Calling static member function without an object
        empCount = Employee::get_NoOfEmplyees();

        cout << "\n\nNumber of Employees currently working in XYZ Company:: "
             << empCount;

        // Adding New Employee
        cout << "\n\nEnter Employee Details: "
             << "\n--------------------------------" << endl;
        emp[empCount].getData();

        cout << "\nEmployee details added successfully!\n\n";

        cout << "Press any key to continue...";
        getch();
        break;

    case 2:

        //Calling friend function to display all records of the Employee
        showEmployeeData(emp);
        cout << "\nPress any key to continue...";
        getch();
        break;
```

```
        case 3:
            return 0;

        default:
            cout << "\nInvalid Choice!";
        }
    }

    return 0;
}


// Member Function
void Employee ::getData(void)
{
    cout << "Employee ID: ";
    cin >> id;

    cout << "Name: ";
    cin >> name;

    cout << "City: ";
    cin >> city;

    cout << "Salary: Rs ";
    cin >> salary;

    NoOfEmployees++;
}


// Member Function
void Employee ::setData(string i, string nam, string cit, float sal)
{
    NoOfEmployees++;
    id = i;
    name = nam;
    city = cit;
    salary = sal;
}


// Member function
void Employee ::setSalary(float money)
{
```

```
    salary = money;
}

float Employee::increaseSalary(float money)
{
    salary = salary + money;

    return salary;
}


// Member Function
void showEmployeeData(Employee *emp)
{
    cout << endl
         << endl
         << right
         << setw(37) << "EMPLOYEE DATA"
         << endl
         << "-----------------------------------------------------------------"
         << endl
         << left
         << setw(10) << "S.No."
         << setw(15) << "Emp ID"
         << setw(15) << "Name"
         << setw(15) << "City"
         << setw(15) << "Salary"
         << endl
         << "-----------------------------------------------------------------"
         << endl
         << endl;

    for (int i = 0, n = emp[0].NoOfEmployees; i < n; i++)
    {

        cout << left
             << setw(10) << i + 1
             << setw(15) << emp[i].id
             << setw(15) << emp[i].name
             << setw(15) << emp[i].city
             << "Rs " << setw(12) << emp[i].salary
             << endl;
    }
    cout << endl;
}
```

OUTPUT:



```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL                                  Code

1. Add Employee
2. Show Employee Data
3. Exit

Enter your choice: 2


                        EMPLOYEE DATA
---------------------------------------------------------------------
S.No.     Emp ID         Name           City           Salary
---------------------------------------------------------------------

1         EMP001         Rohan          Delhi          Rs 50000
2         EMP002         Rahul          Agra           Rs 20000
3         EMP003         Shubham        Lucknow        Rs 25000
4         EMP004         Farhaz         Delhi          Rs 40000
5         EMP005         Anam           Agra           Rs 15000
6         EMP006         Aleena         Agra           Rs 30000
7         EMP007         Anjali         Kanpur         Rs 20000


Press any key to continue...


1. Add Employee
2. Show Employee Data
3. Exit

Enter your choice: 1


Number of Employees currently working in XYZ Company:: 7

Enter Employee Details:
-------------------------------
Employee ID: EMP008
Name: ABC
City: XYZ
Salary: Rs 15000

Employee details added successfully!

Press any key to continue...


1. Add Employee
2. Show Employee Data
3. Exit

Enter your choice: 2
```

OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL Code

```
                    EMPLOYEE DATA
---------------------------------------------------------------------
S.No.     Emp ID        Name          City          Salary
---------------------------------------------------------------------

1         EMP001        Rohan         Delhi         Rs 50000
2         EMP002        Rahul         Agra          Rs 20000
3         EMP003        Shubham       Lucknow       Rs 25000
4         EMP004        Farhaz        Delhi         Rs 40000
5         EMP005        Anam          Agra          Rs 15000
6         EMP006        Aleena        Agra          Rs 30000
7         EMP007        Anjali        Kanpur        Rs 20000
8         EMP008        ABC           XYZ           Rs 15000


Press any key to continue...


1. Add Employee
2. Show Employee Data
3. Exit

Enter your choice: 1


Number of Employees currently working in XYZ Company:: 8

Enter Employee Details:
--------------------------------
Employee ID: EMP009
Name: ABCD1
City: XYZ1
Salary: Rs 25000

Employee details added successfully!

Press any key to continue...


1. Add Employee
2. Show Employee Data
3. Exit

Enter your choice: 2


                    EMPLOYEE DATA
---------------------------------------------------------------------
S.No.     Emp ID        Name          City          Salary
---------------------------------------------------------------------

1         EMP001        Rohan         Delhi         Rs 50000
2         EMP002        Rahul         Agra          Rs 20000
3         EMP003        Shubham       Lucknow       Rs 25000
4         EMP004        Farhaz        Delhi         Rs 40000
5         EMP005        Anam          Agra          Rs 15000
6         EMP006        Aleena        Agra          Rs 30000
7         EMP007        Anjali        Kanpur        Rs 20000
8         EMP008        ABC           XYZ           Rs 15000
9         EMP009        ABCD1         XYZ1          Rs 25000


Press any key to continue...
```

2. ***Write a program to generate results for 10 students using two classes Student and Exam.***

   Student class contains:-
   stud_roll, stud_name, course_name, dob as data members and getrec(), modifyrec(), printrec() as member functions.

   Exam class contains:-
   exam_name, paper_name, paper_code, marks_obtained, total marks, grades as data members and getmarks(), modifymarks(), printresult() as member functions.

SOURCE CODE:

```
#include <iostream>
using namespace std;
#include <iomanip>
#include <string>
#include <conio.h>
#define BLUE "\033[34m"
#define RESET "\033[0m"
#define MAX_STUDENTS 10
#define NO_OF_SUBJECTS 5

class Exam
{
  private:
      string paperName;
      string paperCode;
      float marksObtained;
      float totalMarks;
      string grade;

  public:
      void setExamRecord(string, string, float, float);
      void printExamRecord(int);
      void inputExamRecord();
};

class Student
{
  private:
      string rollNo;
```

```
        string name;
        string courseName;
        string dob;

    public:
        Exam *exam;
        Student();
        ~Student();
        void printResult();
        void inputStudentRecord();
        void setStudentRecord(string, string, string, string);
};

// Parameterized Constructor of Student Class
Student ::Student()
{
    if (NO_OF_SUBJECTS)
        exam = new Exam[NO_OF_SUBJECTS];
    else
        exam = NULL;
}

// Destructor of Student Class
Student::~Student()
{
    delete exam;
}

// Member function of Student Class
void Student ::inputStudentRecord()
{
    cout << "\nEnter Student Details:" << endl
         << "--------------------------------" << endl
         << "Roll No: ";
    cin >> rollNo;
    cout << endl
         << "Name: ";
    cin >> name;
    cout << endl
         << "Course Name: ";
    cin >> courseName;
    cout << "--------------------------------" << endl;
```

```
}

// Member function of Student Class
void Student::setStudentRecord(string rollNo, string name, string courseName,string dob)
{
    this->rollNo = rollNo;
    this->name = name;
    this->courseName = courseName;
    this->dob = dob;
}

// Member function of Student Class
void Student ::printResult()
{
    cout << "  Name          :  " << name << endl
         << "  Roll No       :  " << rollNo << endl
         << "  Course Name   :  " << courseName << endl
         << "  Date of Birth :  " << dob << endl
         << endl
         << "-------------------------------------------------------------"
         << "----------------------------------------" << endl
         << "   " << left << setw(11) << "S.No."
         << setw(19) << "Paper Name"
         << setw(19) << "Paper Code"
         << setw(23) << "Marks Obtained"
         << setw(19) << "Total Marks"
         << setw(19) << "Grade" << endl
         << "-------------------------------------------------------------"
         << "----------------------------------------" << endl

    for (int i = 0; i < NO_OF_SUBJECTS; i++)
    {
        exam[i].printExamRecord(i);
    }

    cout << endl
         << endl;
}

// Member function of Exam Class
void Exam ::inputExamRecord()
{
```

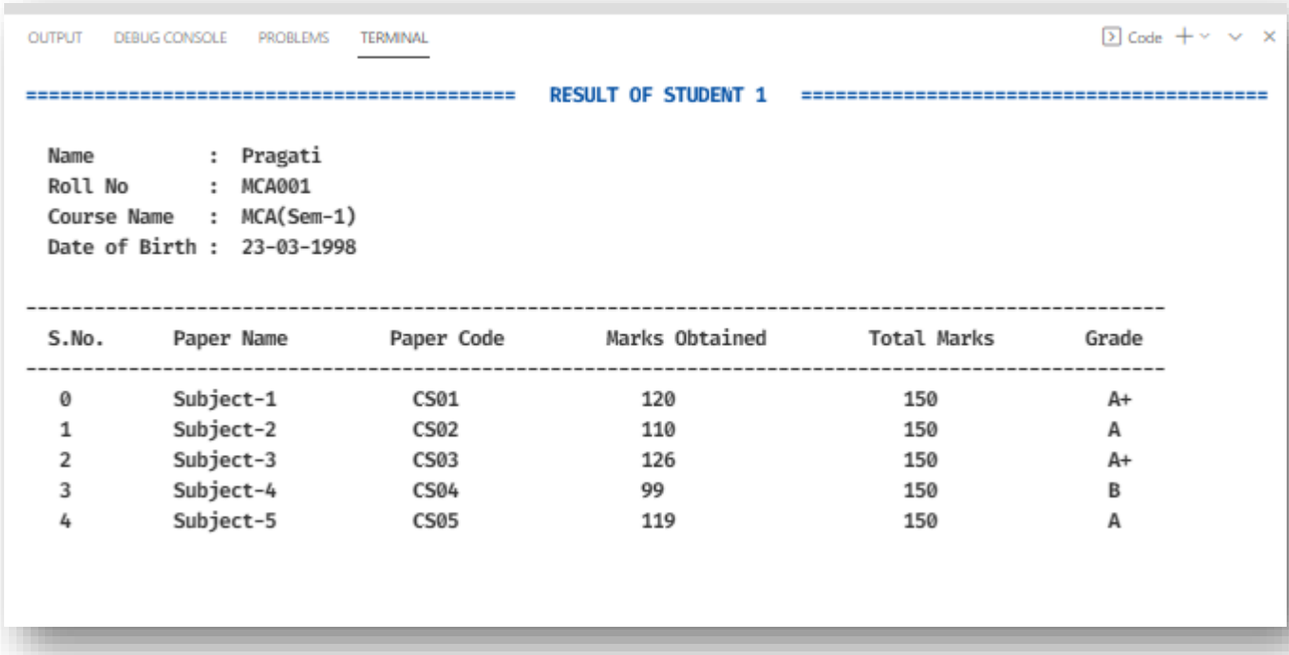
```
    cout << "\nEnter Exam Details:" << endl
         << "--------------------------------" << endl
         << "Paper Name: ";
    cin >> paperName;
    cout << endl
         << "Paper Code: ";
    cin >> paperCode;
    cout << endl
         << "Marks Obtained: ";
    cin >> marksObtained;
    cout << endl
         << "Total Marks: ";
    cin >> totalMarks;
    cout << "--------------------------------" << endl;
}

// Member function of Exam Class
void Exam ::setExamRecord(string paperName, string paperCode, float marksObtained
, float totalMarks)
{
    this->paperName = paperName;
    this->paperCode = paperCode;
    this->marksObtained = marksObtained;
    this->totalMarks = totalMarks;

    float percentage = marksObtained / totalMarks * 100;

    if (percentage >= 90.0)
        this->grade = "A++";
    else if (percentage < 90.0 && percentage >= 80.0)
        this->grade = "A+";
    else if (percentage < 80.0 && percentage >= 70.0)
        this->grade = "A";
    else if (percentage < 70.0 && percentage >= 60.0)
        this->grade = "B";
    else if (percentage < 60.0 && percentage >= 50.0)
        this->grade = "C";
    else if (percentage < 50.0 && percentage >= 40.0)
        this->grade = "D";
    else
        this->grade = "F";
}
```

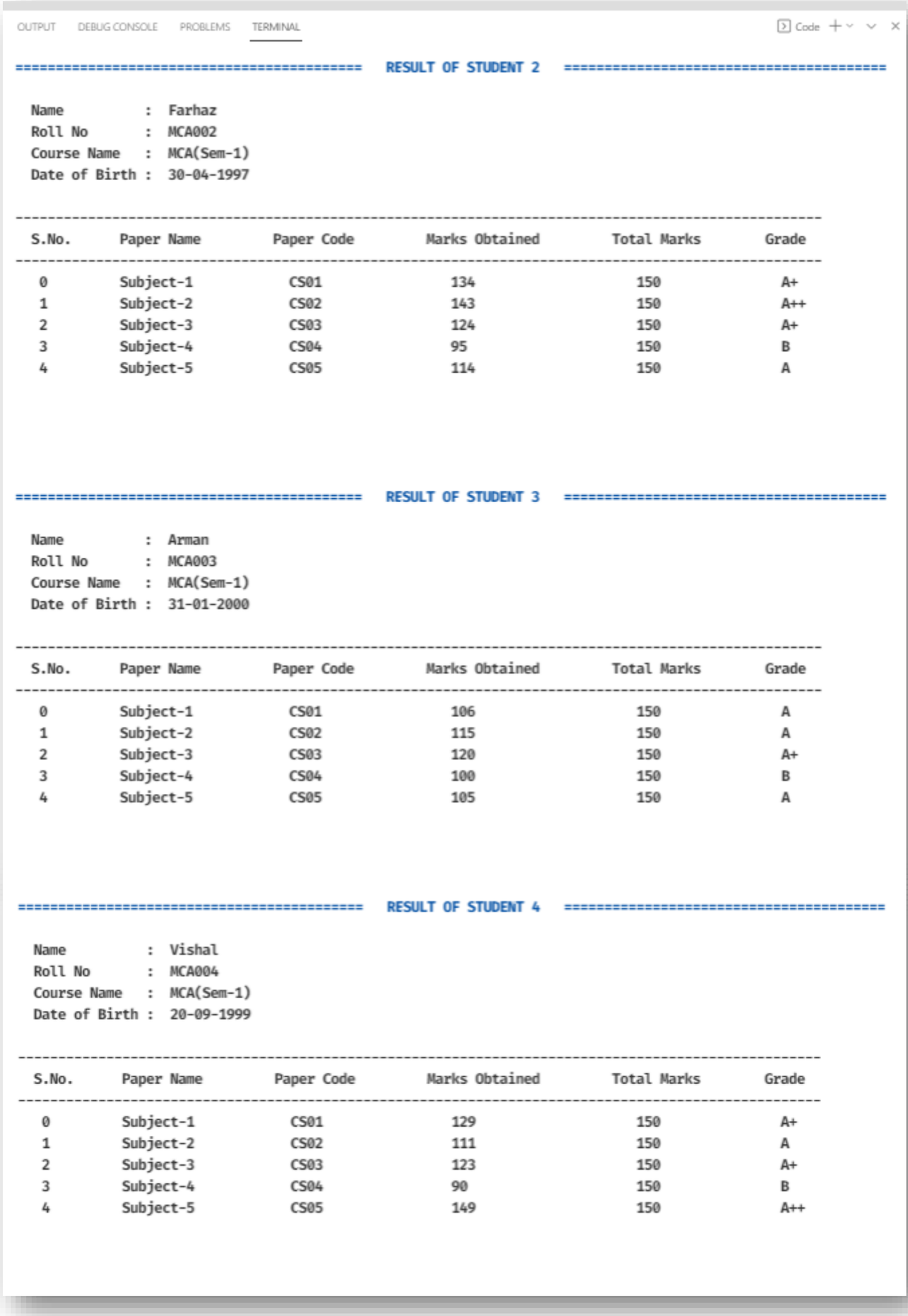
```
// Member function of Exam class
void Exam ::printExamRecord(int i)
{
    cout << "    " << left << setw(10) << i
         << setw(19) << paperName
         << "   " << setw(17) << paperCode
         << "    " << setw(20) << marksObtained
         << "    " << setw(16) << totalMarks
         << "   " << setw(17) << grade << endl;
}

// Driver Code
int main(void)
{
    Student stud[MAX_STUDENTS];

    // Student's Record (10 Students)

    stud[0].setStudentRecord("MCA001", "Pragati", "MCA(Sem-1)", "23-03-1998");
    stud[1].setStudentRecord("MCA002", "Farhaz", "MCA(Sem-1)", "30-04-1997");
    stud[2].setStudentRecord("MCA003", "Arman", "MCA(Sem-1)", "31-01-2000");
    stud[3].setStudentRecord("MCA004", "Vishal", "MCA(Sem-1)", "20-09-1999");
    stud[4].setStudentRecord("MCA005", "Aleena", "MCA(Sem-1)", "25-02-1996");
    stud[5].setStudentRecord("MCA006", "Anjali", "MCA(Sem-1)", "19-05-1995");
    stud[6].setStudentRecord("MCA007", "Akanksha", "MCA(Sem-1)", "01-11-1999");
    stud[7].setStudentRecord("MCA008", "Rahul", "MCA(Sem-1)", "12-08-1998");
    stud[8].setStudentRecord("MCA009", "Karan", "MCA(Sem-1)", "28-01-1999");
    stud[9].setStudentRecord("MCA009", "Arzoo", "MCA(Sem-1)", "28-06-1999");

    // Student's Exam Record (5 Subjects)

    stud[0].exam[0].setExamRecord("Subject-1", "CS01", 120, 150);
    stud[0].exam[1].setExamRecord("Subject-2", "CS02", 110, 150);
    stud[0].exam[2].setExamRecord("Subject-3", "CS03", 126, 150);
    stud[0].exam[3].setExamRecord("Subject-4", "CS04", 99, 150);
    stud[0].exam[4].setExamRecord("Subject-5", "CS05", 119, 150);

    stud[1].exam[0].setExamRecord("Subject-1", "CS01", 134, 150);
    stud[1].exam[1].setExamRecord("Subject-2", "CS02", 143, 150);
    stud[1].exam[2].setExamRecord("Subject-3", "CS03", 124, 150);
    stud[1].exam[3].setExamRecord("Subject-4", "CS04", 95, 150);
    stud[1].exam[4].setExamRecord("Subject-5", "CS05", 114, 150);
```

```
stud[2].exam[0].setExamRecord("Subject-1", "CS01", 106, 150);
stud[2].exam[1].setExamRecord("Subject-2", "CS02", 115, 150);
stud[2].exam[2].setExamRecord("Subject-3", "CS03", 120, 150);
stud[2].exam[3].setExamRecord("Subject-4", "CS04", 100, 150);
stud[2].exam[4].setExamRecord("Subject-5", "CS05", 105, 150);

stud[3].exam[0].setExamRecord("Subject-1", "CS01", 129, 150);
stud[3].exam[1].setExamRecord("Subject-2", "CS02", 111, 150);
stud[3].exam[2].setExamRecord("Subject-3", "CS03", 123, 150);
stud[3].exam[3].setExamRecord("Subject-4", "CS04", 90, 150);
stud[3].exam[4].setExamRecord("Subject-5", "CS05", 149, 150);

stud[4].exam[0].setExamRecord("Subject-1", "CS01", 120, 150);
stud[4].exam[1].setExamRecord("Subject-2", "CS02", 110, 150);
stud[4].exam[2].setExamRecord("Subject-3", "CS03", 116, 150);
stud[4].exam[3].setExamRecord("Subject-4", "CS04", 95, 150);
stud[4].exam[4].setExamRecord("Subject-5", "CS05", 132, 150);

stud[5].exam[0].setExamRecord("Subject-1", "CS01", 124, 150);
stud[5].exam[1].setExamRecord("Subject-2", "CS02", 105, 150);
stud[5].exam[2].setExamRecord("Subject-3", "CS03", 122, 150);
stud[5].exam[3].setExamRecord("Subject-4", "CS04", 90, 150);
stud[5].exam[4].setExamRecord("Subject-5", "CS05", 109, 150);

stud[6].exam[0].setExamRecord("Subject-1", "CS01", 120, 150);
stud[6].exam[1].setExamRecord("Subject-2", "CS02", 110, 150);
stud[6].exam[2].setExamRecord("Subject-3", "CS03", 126, 150);
stud[6].exam[3].setExamRecord("Subject-4", "CS04", 99, 150);
stud[6].exam[4].setExamRecord("Subject-5", "CS05", 119, 150);

stud[7].exam[0].setExamRecord("Subject-1", "CS01", 120, 150);
stud[7].exam[1].setExamRecord("Subject-2", "CS02", 110, 150);
stud[7].exam[2].setExamRecord("Subject-3", "CS03", 126, 150);
stud[7].exam[3].setExamRecord("Subject-4", "CS04", 99, 150);
stud[7].exam[4].setExamRecord("Subject-5", "CS05", 119, 150);

stud[8].exam[0].setExamRecord("Subject-1", "CS01", 120, 150);
stud[8].exam[1].setExamRecord("Subject-2", "CS02", 110, 150);
stud[8].exam[2].setExamRecord("Subject-3", "CS03", 126, 150);
stud[8].exam[3].setExamRecord("Subject-4", "CS04", 99, 150);
stud[8].exam[4].setExamRecord("Subject-5", "CS05", 119, 150);
```

```
    stud[9].exam[0].setExamRecord("Subject-1", "CS01", 120, 150);
    stud[9].exam[1].setExamRecord("Subject-2", "CS02", 110, 150);
    stud[9].exam[2].setExamRecord("Subject-3", "CS03", 126, 150);
    stud[9].exam[3].setExamRecord("Subject-4", "CS04", 99, 150);
    stud[9].exam[4].setExamRecord("Subject-5", "CS05", 119, 150);

    // Printing Results of Students
    for (int i = 0; i < MAX_STUDENTS; i++)
    {

        cout << endl
             << endl
             << endl
             << BLUE
             << "=========================================    "
             << "RESULT OF STUDENT " << (i + 1)
             << "    ========================================="
             << RESET
             << endl
             << endl;

        stud[i].printResult();
    }

    return 0;
}
```

OUTPUT:




OUTPUT
DEBUG CONSOLE
PROBLEMS
TERMINAL
Code
========== RESULT OF STUDENT 2 ==========
Name : Farhaz
Roll No : MCA002
Course Name : MCA(Sem-1)
Date of Birth : 30-04-1997
S.No. Paper Name Paper Code Marks Obtained Total Marks Grade
0 Subject-1 CS01 134 150 A+
1 Subject-2 CS02 143 150 A++
2 Subject-3 CS03 124 150 A+
3 Subject-4 CS04 95 150 B
4 Subject-5 CS05 114 150 A
========== RESULT OF STUDENT 3 ==========
Name : Arman
Roll No : MCA003
Course Name : MCA(Sem-1)
Date of Birth : 31-01-2000
S.No. Paper Name Paper Code Marks Obtained Total Marks Grade
0 Subject-1 CS01 106 150 A
1 Subject-2 CS02 115 150 A
2 Subject-3 CS03 120 150 A+
3 Subject-4 CS04 100 150 B
4 Subject-5 CS05 105 150 A
========== RESULT OF STUDENT 4 ==========
Name : Vishal
Roll No : MCA004
Course Name : MCA(Sem-1)
Date of Birth : 20-09-1999
S.No. Paper Name Paper Code Marks Obtained Total Marks Grade
0 Subject-1 CS01 129 150 A+
1 Subject-2 CS02 111 150 A
2 Subject-3 CS03 123 150 A+
3 Subject-4 CS04 90 150 B
4 Subject-5 CS05 149 150 A++

3. ***Write a program to implement the member functions of a Class Shape having the same name, calculate_area() for calculating the area of a Triangle, Rectangle and Circle using the concept of Function overloading.***

SOURCE CODE:

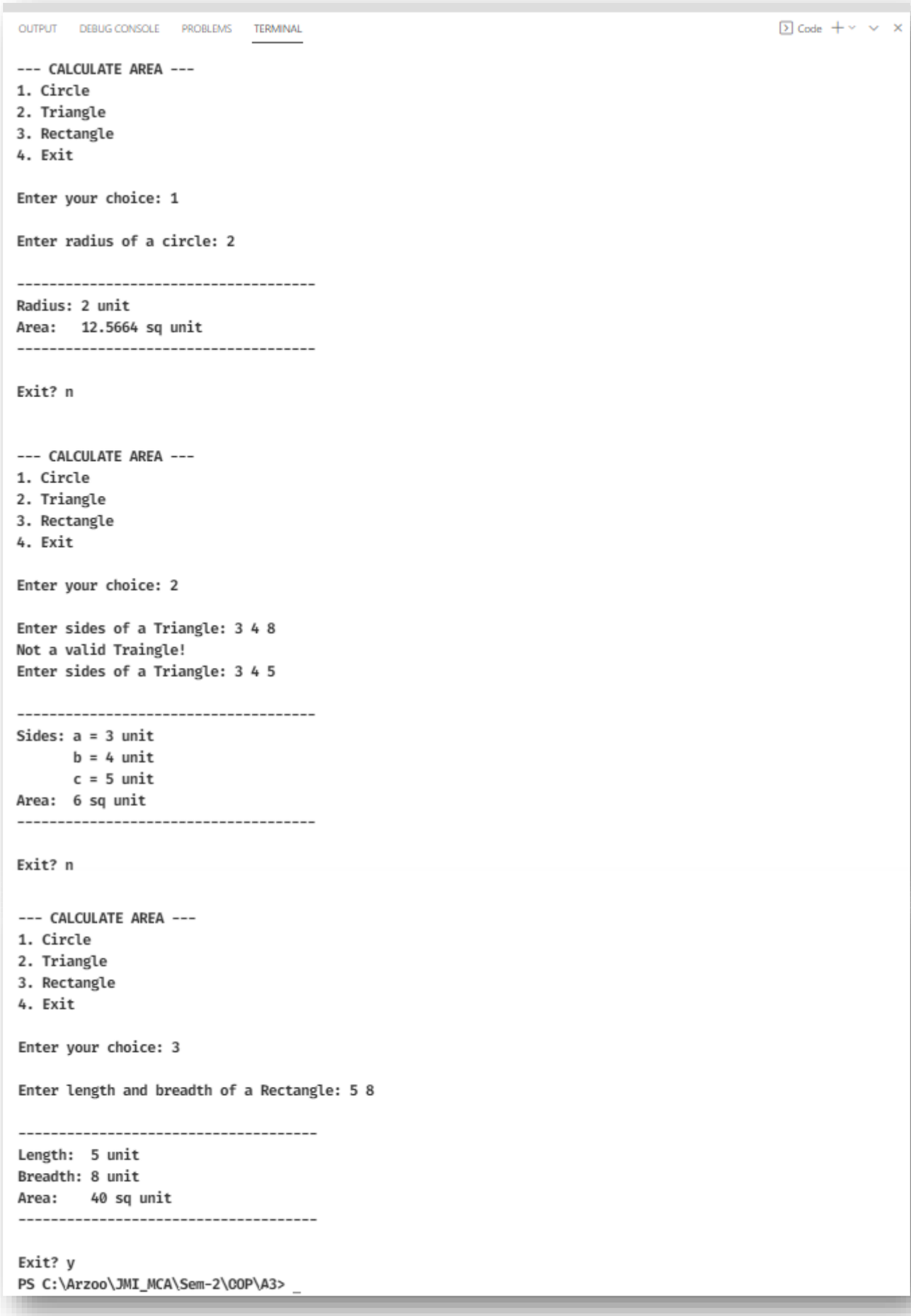
```
#include <iostream>
using namespace std;
#include <conio.h>
#include <cstdbool>
#include <cmath>
#define PI 3.14159

// Function overloading
float area(float);               // Circle
float area(float, float);        // Rectangle
float area(float, float, float); // Triangle


// Driver Code
int main(void)
{
    int choice;
    char Exit;

    do
    {
        cout << "\n\n--- CALCULATE AREA ---" << endl
             << "1. Circle" << endl
             << "2. Triangle" << endl
             << "3. Rectangle" << endl
             << "4. Exit" << endl
             << "\nEnter your choice: ";
        cin >> choice;

        switch (choice)
        {
        case 1:

            float radius;
            bool isCircle;

            // This loop will not terminate until user enters a valid radius
            do
            {
```

```
            isCircle = false;

            // Input radius of the circle from the user
            cout << "\nEnter radius of a circle: ";
            cin >> radius;

            // A valid radius would be a positive number
            if (radius >= 0)
            {
                isCircle = true;
            }
            else
            {
                cout << "Not a valid Radius!";
            }

        } while (isCircle == false);

        cout << "\n-------------------------------------\n"
             << "Radius: " << radius << " unit" << endl
             << "Area:   " << area(radius) << " sq unit" //Function Call
             << "\n-------------------------------------\n";
        break;


    case 2:

        float a, b, c;
        bool isTriangle;

        // This loop will not terminate until user enters the valid
        // sides of a triangle
        do
        {
            isTriangle = false;

            // Input sides of the triangle from the user
            cout << "\nEnter sides of a Triangle: ";
            cin >> a >> b >> c;

            // Conditions for the length of sides to be a valid triangle
            if (a>=0 && b>=0 && c>=0 && a+b > c && b+c > a && c+a > b)
            {
                isTriangle = true;
            }
            else
```

```
        {
            cout << "Not a valid Traingle!";
        }

    } while (isTriangle == false);

    cout << "\n-------------------------------------\n"
         << "Sides: a = " << a << " unit" << endl
         << "       b = " << b << " unit" << endl
         << "       c = " << c << " unit" << endl
         << "Area:  " << area(a, b, c) << " sq unit" //Function Call
         << "\n-------------------------------------\n";

    break;

case 3:

    float length, breadth;
    bool isRectangle;

    // This loop will not terminate until user enters the valid
    // length and breadth of a rectangle
    do
    {
        isRectangle = false;

        // Input length and breadth of the rectangle from the user
        cout << "\nEnter length and breadth of a Rectangle: ";
        cin >> length >> breadth;

        // The length and breadth of the rectangle has to be
        // positive number
        if (length >= 0 || breadth >= 0)
        {
            isRectangle = true;
        }
        else
        {
            cout << "Not a valid Rectangle!";
        }

    } while (isRectangle == false);

    cout << "\n-------------------------------------\n"
         << "Length:  " << length << " unit" << endl
         << "Breadth: " << breadth << " unit" << endl
```

```
                << "Area:    " << area(length, breadth) << " sq unit"
                << "\n-------------------------------------\n";

            break;

        case 4:
            return 0;

        default:
            cout << "\nInvalid choice!";
        }

        cout << "\nExit? ";
        cin >> Exit;
    } while (Exit != 'y' && Exit != 'Y');

    return 0;
}

// Function to calculate area of a Circle
float area(float radius)
{
    return (PI * radius * radius);
}

// Function to calculate area of a Rectangle
float area(float length, float breadth)
{
    return (length * breadth);
}

// Function to calcuate area of a Triangle
float area(float a, float b, float c)
{
    /*
         Heron's Formula
         Area => Δ =  √(s(s-a)(s-b)(s-c))
         Semiperimeter => s = (a + b + c) / 2
    */

    float s = (a + b + c) / 2;
    return (sqrt(s * (s - a) * (s - b) * (s - c)));
}
```

OUTPUT:



```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL                                  Code

--- CALCULATE AREA ---
1. Circle
2. Triangle
3. Rectangle
4. Exit

Enter your choice: 1

Enter radius of a circle: 2

--------------------------------------
Radius: 2 unit
Area:   12.5664 sq unit
--------------------------------------

Exit? n


--- CALCULATE AREA ---
1. Circle
2. Triangle
3. Rectangle
4. Exit

Enter your choice: 2

Enter sides of a Triangle: 3 4 8
Not a valid Traingle!
Enter sides of a Triangle: 3 4 5

--------------------------------------
Sides: a = 3 unit
       b = 4 unit
       c = 5 unit
Area:  6 sq unit
--------------------------------------

Exit? n


--- CALCULATE AREA ---
1. Circle
2. Triangle
3. Rectangle
4. Exit

Enter your choice: 3

Enter length and breadth of a Rectangle: 5 8

--------------------------------------
Length:  5 unit
Breadth: 8 unit
Area:    40 sq unit
--------------------------------------

Exit? y
PS C:\Arzoo\JMI_MCA\Sem-2\COP\A3> _
```

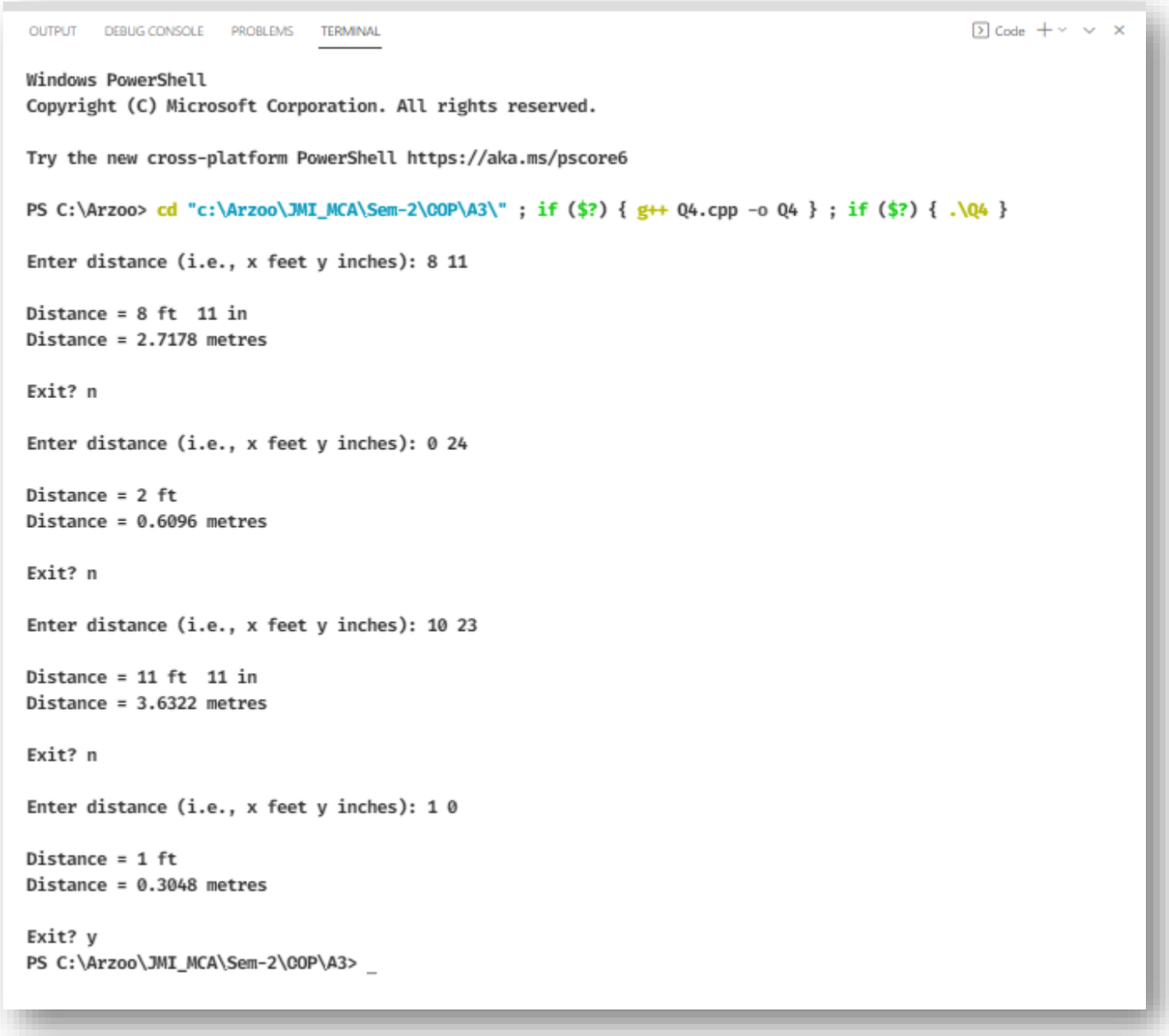
4. ***Write a program to convert a distance entered in Feet and Inches to Meters using class to basic data type conversion.***

SOURCE CODE:

```
#include <iostream>
using namespace std;
#define FOOT_TO_METRE 0.3048

class Distance
{
    int foot;
    int inch;

public:
    operator float();
    friend istream& operator>>(istream &, Distance &);
    friend ostream& operator<<(ostream &, Distance &);
};


// Casting Operator (Class to Basic Data Type Conversion)
Distance::operator float()
{
    float metre = FOOT_TO_METRE * (foot + inch / 12.0);

    return metre;
}


// '>>' Opeartor Overloading
istream& operator>>(istream &Cin, Distance &d)
{
    cin >> d.foot >> d.inch;

    if (d.inch >= 12)
    {
        d.foot += d.inch / 12;
        d.inch %= 12;
    }
    return Cin;
}


// '<<' Opeartor Overloading
```

```
ostream& operator<<(ostream &Cout, Distance &d)
{
    if (d.foot != 0)
        cout << d.foot << " ft  ";
    if (d.inch != 0)
        cout << d.inch << " in";

    return Cout;
}


// Driver Code
int main(void)
{
    Distance d1;
    float metre;
    char Exit;

    do
    {
        // Input distance from user in feet and inches
        cout << "\nEnter distance (i.e., x feet y inches): ";
        cin >> d1;


        metre = d1; // Class to Basic data type conversion


        // Display distance
        cout << "\nDistance = " << d1
             << "\nDistance = " << metre << " metres\n\n";


        cout << "Exit? ";
        cin >> Exit;

    } while (Exit != 'y' && Exit != 'Y');

    return 0;
}
```

OUTPUT:


OUTPUT
DEBUG CONSOLE
PROBLEMS
TERMINAL
Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Arzoo> cd "c:\Arzoo\JMI_MCA\Sem-2\COP\A3\" ; if ($?) { g++ Q4.cpp -o Q4 } ; if ($?) { .\Q4 }
Enter distance (i.e., x feet y inches): 8 11
Distance = 8 ft 11 in
Distance = 2.7178 metres
Exit? n
Enter distance (i.e., x feet y inches): 0 24
Distance = 2 ft
Distance = 0.6096 metres
Exit? n
Enter distance (i.e., x feet y inches): 10 23
Distance = 11 ft 11 in
Distance = 3.6322 metres
Exit? n
Enter distance (i.e., x feet y inches): 1 0
Distance = 1 ft
Distance = 0.3048 metres
Exit? y
PS C:\Arzoo\JMI_MCA\Sem-2\COP\A3> _

*****