OBJECT ORIENTED PROGRAMING

OOP

PRACTICALS

NAME: SYEDA FAIZA ASLAM

SECTION: B

ROLL NO: CT-25064

LAB NO: 02

COURSE CODE: CT-260

COURSE TEACHER: Mr. AHMED ZAKI

Exercise

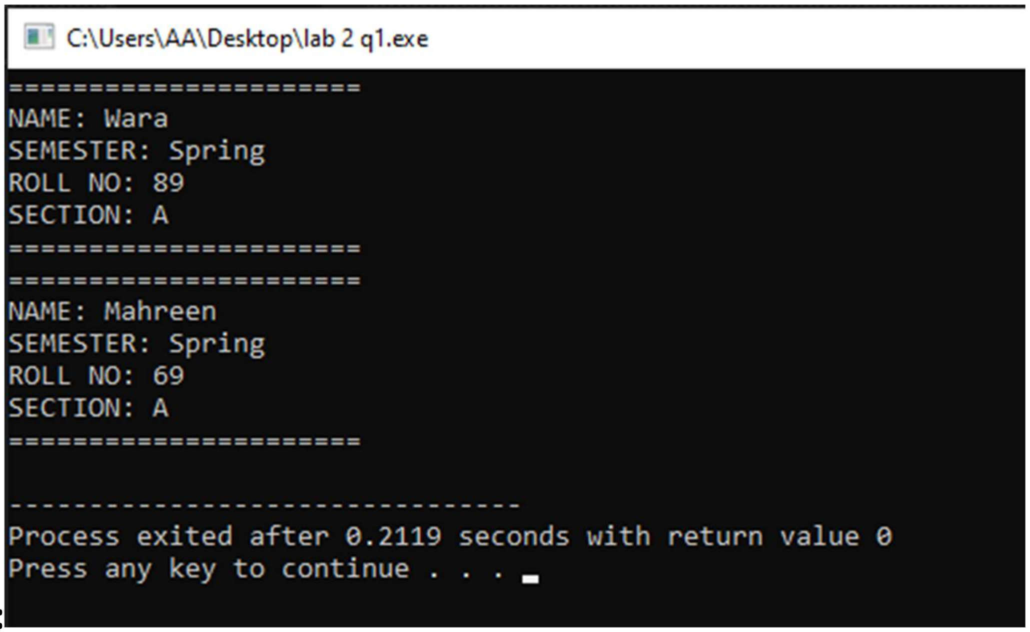
1. Write a program in which a class named student has member variables name, roll no, semester and section. Create 4 objects of this class to store data of 4 different students, now display data of only those students who belong to section A.

SOURCE CODE:

```
#include<iostream>
using namespace std;
class Student
{
	private:
		string name;
		string semester;
		int roll_no;
		char sec;
	public:
		void displayData()
		{
			if( sec == 'A')
			{
				cout<<"======================\n";
				cout<<"NAME: "<<name<<endl;
				cout<<"SEMESTER: "<<semester<<endl;
				cout<<"ROLL NO: "<<roll_no<<endl;
				cout<<"SECTION: "<<sec<<endl;
				cout<<"======================\n";
			}
		}
		string getName()
		{
			return name;
		}
		string getSemester()
		{
			return semester;
		}
		int getRollNo()
		{
			return roll_no;
		}
		char getSection()
		{
			return sec;
		}
		void setName(string name)
		{
			this->name = name;
		}
		void setSemester(string semester)
		{
			this->semester = semester;
		}
		void setRollNo(int roll_no)
		{
			this->roll_no = roll_no;
		}
		void setSection(char sec)
		{
			this->sec = sec;
		}
};
int main()
{
	Student s1,s2,s3,s4;
	s1.setName("Faiza");
	s1.setSemester("Fall");
```

```
        s1.setRollNo(64);
        s1.setSection('B');
        s1.displayData();

        s2.setName("Wara");
        s2.setSemester("Spring");
        s2.setRollNo(89);
        s2.setSection('A');
        s2.displayData();

        s3.setName("Amaan");
        s3.setSemester("Fall");
        s3.setRollNo(77);
        s3.setSection('B');
        s3.displayData();

        s4.setName("Mahreen");
        s4.setSemester("Spring");
        s4.setRollNo(69);
        s4.setSection('A');
        s4.displayData();

        return 0;
}
```

OUTPUT


C:\Users\AA\Desktop\lab 2 q1.exe

```
======================
NAME: Wara
SEMESTER: Spring
ROLL NO: 89
SECTION: A
======================
======================
NAME: Mahreen
SEMESTER: Spring
ROLL NO: 69
SECTION: A
======================


--------------------------------
Process exited after 0.2119 seconds with return value 0
Press any key to continue . . .
```

:

2. You are a programmer for the ABC Bank assigned to develop a class that models the basic workings of a bank account. The class should perform the following tasks:

o Save the account balance.

o Save the number of transactions performed on the account.

o Allow deposits to be made to the account.

o Allow with drawls to be taken from the account.

o Report the current account balance at any time.

o Report the current number of transactions at any time.

Menu

1. Display the account balance

2. Display the number of transactions

3. Display interest earned for this period

4. Make a deposit

5. Make a withdrawal

6. Exit the program

SOURCE CODE:

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

class BankAccount
{
private:
   float balance;
   int transaction;
   float interest_Rate;
```

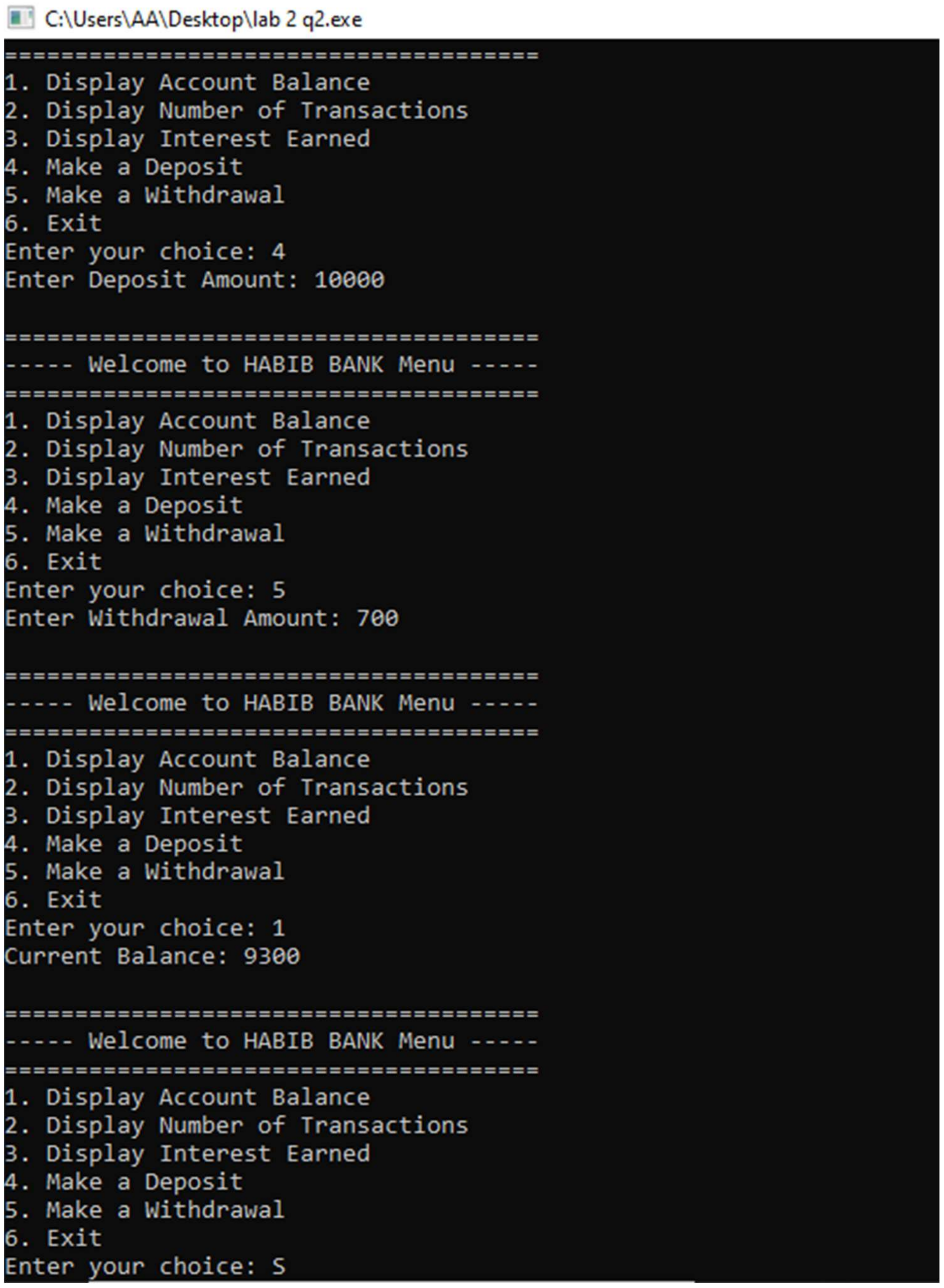
```
public:
  BankAccount()
  {
    balance = 0;
    transaction = 0;
    interest_Rate = 0.05;
  }

  void deposit(double amount)
  {
    if (amount > 0)
    {
      balance += amount;
      transaction++;
    }
    else
    {
      cout << "Invalid amount!\n";
    }
  }

  void withdraw(double amount)
  {
    if (amount > 0 && amount <= balance)
    {
      balance -= amount;
      transaction++;
    }
    else
    {
      cout << "Invalid amount or insufficient balance!\n";
    }
  }
```

```
    double getBalance()
    {
        return balance;
    }

    int getTransaction()
    {
        return transaction;
    }

    double calculateInterest()
    {
        return balance * interest_Rate;
    }
};

int main()
{
    BankAccount acc;
    int choice;
    double amount;

    do
    {
        cout << "\n========================================\n";
        cout << "----- Welcome to HABIB BANK Menu -----\n";
        cout << "========================================\n";
        cout << "1. Display Account Balance\n";
        cout << "2. Display Number of Transactions\n";
        cout << "3. Display Interest Earned\n";
        cout << "4. Make a Deposit\n";
        cout << "5. Make a Withdrawal\n";
        cout << "6. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
```

```
switch (choice)
{
case 1:
   cout << "Current Balance: " << acc.getBalance() << endl;
   break;

case 2:
   cout << "Number of Transactions: "<< acc.getTransaction() << endl;
   break;

case 3:
   cout << "Interest Earned: "<< acc.calculateInterest() << endl;
   break;

case 4:
   cout << "Enter Deposit Amount: ";
   cin >> amount;
   acc.deposit(amount);
   break;

case 5:
   cout << "Enter Withdrawal Amount: ";
   cin >> amount;
   acc.withdraw(amount);
   break;

case 6:
   cout << "Thank you for visiting HABIB BANK!\n";
   break;

default:
   cout << "Invalid choice! Try again.\n";
}
```

```
    } while (choice != 6);

    return 0;
}
```

OUTPUT:



C:\Users\AA\Desktop\lab 2 q2.exe

```
======================================
1. Display Account Balance
2. Display Number of Transactions
3. Display Interest Earned
4. Make a Deposit
5. Make a Withdrawal
6. Exit
Enter your choice: 4
Enter Deposit Amount: 10000

======================================
----- Welcome to HABIB BANK Menu -----
======================================
1. Display Account Balance
2. Display Number of Transactions
3. Display Interest Earned
4. Make a Deposit
5. Make a Withdrawal
6. Exit
Enter your choice: 5
Enter Withdrawal Amount: 700

======================================
----- Welcome to HABIB BANK Menu -----
======================================
1. Display Account Balance
2. Display Number of Transactions
3. Display Interest Earned
4. Make a Deposit
5. Make a Withdrawal
6. Exit
Enter your choice: 1
Current Balance: 9300

======================================
----- Welcome to HABIB BANK Menu -----
======================================
1. Display Account Balance
2. Display Number of Transactions
3. Display Interest Earned
4. Make a Deposit
5. Make a Withdrawal
6. Exit
Enter your choice: S
```

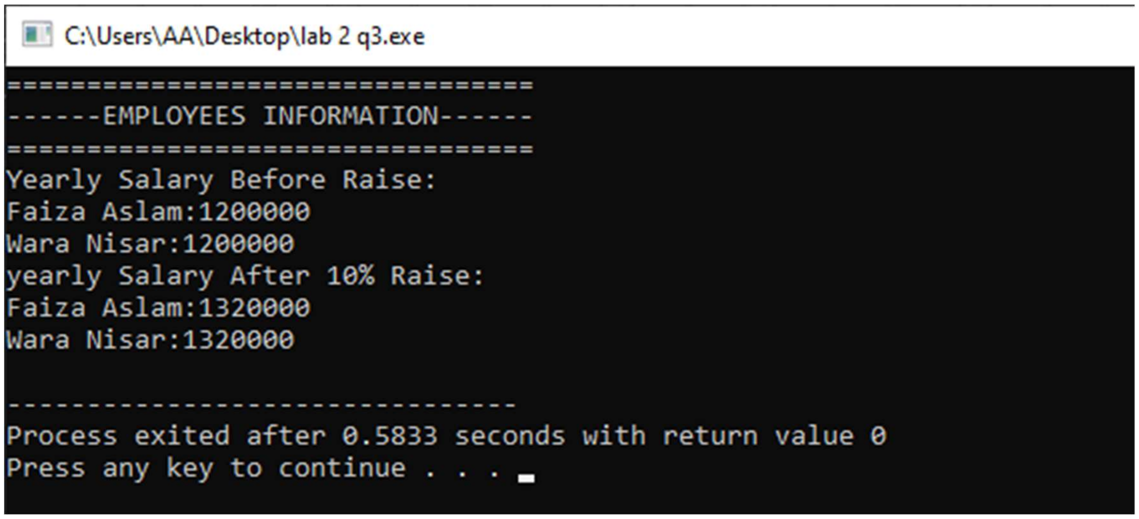
<u>3. Create a class called Employee that includes three pieces of information as data members—a first name (type char* (DMA)), a last name (type string) and a monthly salary (type int). Your class should have a setter function that initializes the three data members. Provide a getter function for each data member. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again. Identify and add any other related functions to achieve the said goal.</u>

<u>SOURCE CODE:</u>

```
#include<iostream>
#include <string>
#include <cstring>
using namespace std;
class Employee
{
        private:
                char* firstName;
                string lastName;
                int monthlySalary;
        public:
                Employee()
                {
                        firstName = nullptr;
                        monthlySalary = 0;
                }
                void setEmployeeData(const char* f_name, string l_name, int salary)
                {
                        firstName = new char [strlen(f_name)+1];
                        strcpy(firstName,f_name);
                        lastName = l_name;
```

```
			if(salary>0)
			{
				monthlySalary = salary;
			}
			else
				monthlySalary = 0;
		}
		char* getFirstName()
		{
			return firstName;
		}
		string getLastName()
		{
			return lastName;
		}
		int getMonthlySalary()
		{
			return monthlySalary;
		}
		int getYearlySalary()
		{
			return monthlySalary * 12;
		}
		void giveRaise()
		{
			monthlySalary = monthlySalary + (monthlySalary*0.10);
		}
		~Employee()
		{
			delete[]firstName;
		}
};
int main()
{
	Employee e1, e2;
	cout<<"==================================\n";
	cout<<"------EMPLOYEES INFORMATION------\n";
	cout<<"==================================\n";
	e1.setEmployeeData("Faiza","Aslam",100000);
	e2.setEmployeeData("Wara","Nisar",100000);
	cout<<"Yearly Salary Before Raise:\n";
	cout<<e1.getFirstName()<<""<<e1.getLastName()<<":"<<e1.getYearlySalary
()<<endl;
	cout<<e2.getFirstName()<<""<<e2.getLastName()<<":"<<e2.getYearlySalary
()<<endl;
	e1.giveRaise();
	e2.giveRaise();
	cout<<"yearly Salary After 10% Raise:\n";
	cout<<e1.getFirstName()<<"
"<<e1.getLastName()<<":"<<e1.getYearlySalary()<<endl;
	cout<<e2.getFirstName()<<"
"<<e2.getLastName()<<":"<<e2.getYearlySalary()<<endl;
	return 0;
}
```

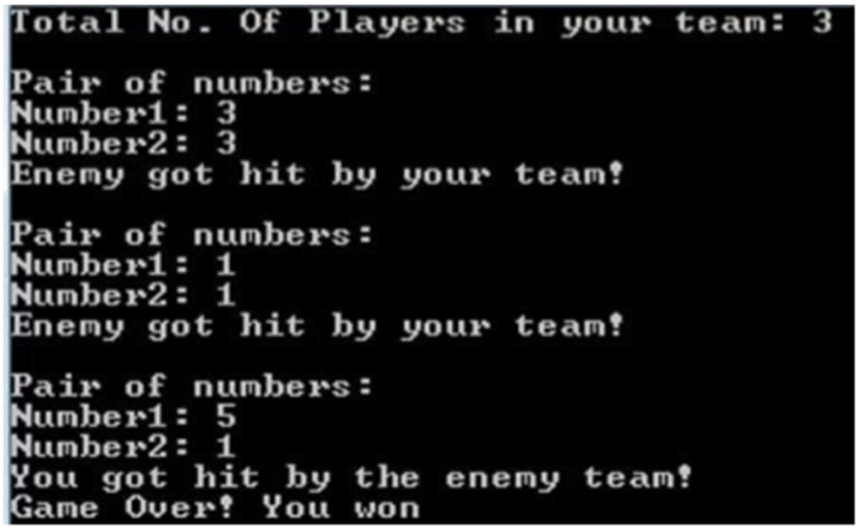
OUTPUT:



C:\Users\AA\Desktop\lab 2 q3.exe

```
=================================
------EMPLOYEES INFORMATION------
=================================
Yearly Salary Before Raise:
Faiza Aslam:1200000
Wara Nisar:1200000
yearly Salary After 10% Raise:
Faiza Aslam:1320000
Wara Nisar:1320000

--------------------------------
Process exited after 0.5833 seconds with return value 0
Press any key to continue . . .
```

4. Write C++ code to represent a hitting game by using OOP concept. The details are as follows: This game is being played between two teams (i.e. your team and the enemy team). The total number of players in your team is randomly generated and stored accordingly. The function generates a pair of numbers and matches each pair. If the numbers get matched, the following message is displayed: "Enemy got hit by your team!" Otherwise, the following message is displayed: "You got hit by the enemy team!" The number of hits should be equal to the number of players in your team. The program should tell the final result of your team by counting



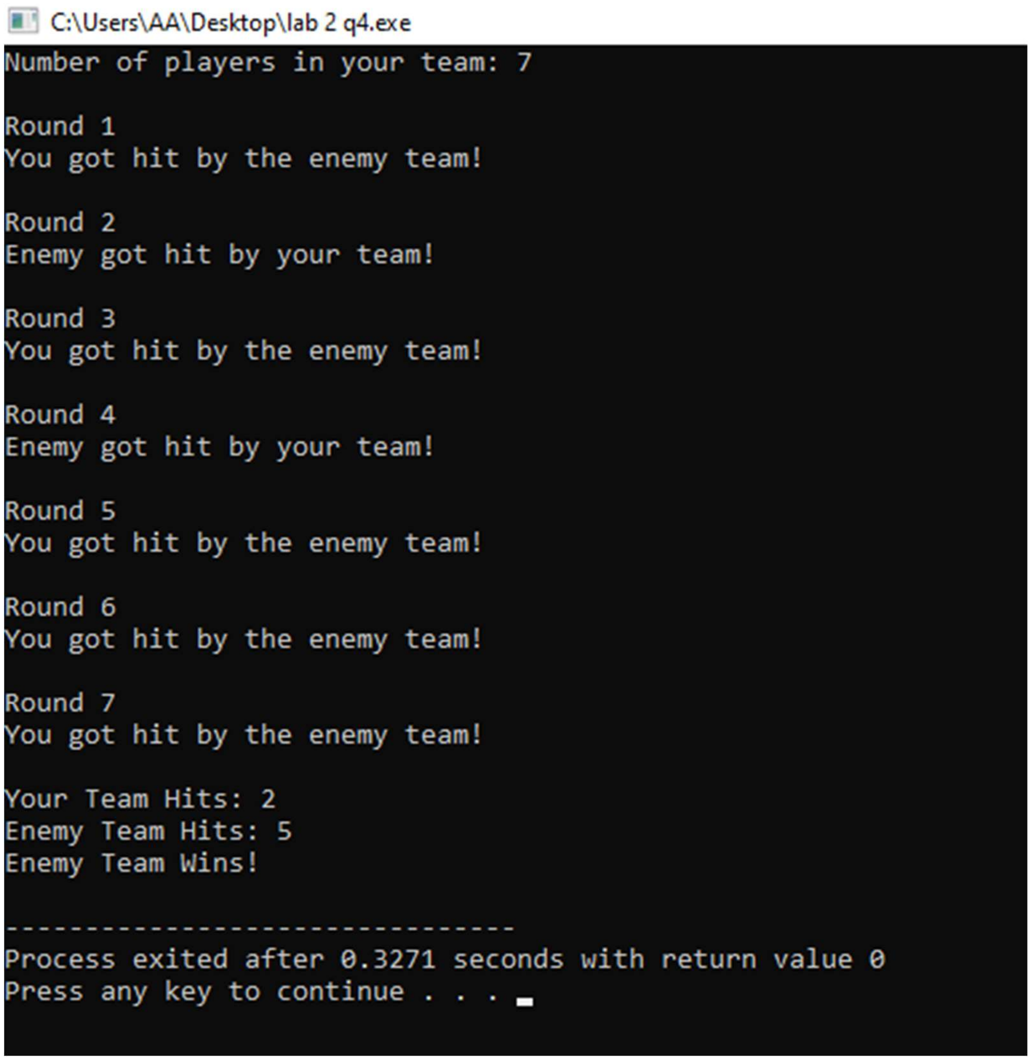
```
Total No. Of Players in your team: 3

Pair of numbers:
Number1: 3
Number2: 3
Enemy got hit by your team!

Pair of numbers:
Number1: 1
Number2: 1
Enemy got hit by your team!

Pair of numbers:
Number1: 5
Number2: 1
You got hit by the enemy team!
Game Over! You won
```

SOURCE CODE:

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

class HittingGame
{
private:
   int players;
   int yourHits;
```

```
    int enemyHits;

public:
    HittingGame()
    {
        yourHits = 0;
        enemyHits = 0;
        srand(time(0));
        players = rand() % 10 + 1;
    }

    void playGame()
    {
        cout << "Number of players in your team: " << players << endl << endl;

        for (int i = 1; i <= players; i++)
        {
            int yourNum = rand() % 5 + 1;
            int enemyNum = rand() % 5 + 1;

            cout << "Round " << i << endl;

            if (yourNum == enemyNum)
            {
                cout << "Enemy got hit by your team!" << endl << endl;
                yourHits++;
            }
            else
            {
                cout << "You got hit by the enemy team!" << endl << endl;
                enemyHits++;
            }
        }
    }
    void showResult()
    {
        cout << "Your Team Hits: " << yourHits << endl;
        cout << "Enemy Team Hits: " << enemyHits << endl;

        if (yourHits > enemyHits)
            cout << "Your Team Wins!" << endl;
        else if (enemyHits > yourHits)
            cout << "Enemy Team Wins!" << endl;
        else
            cout << "Match Draw!" << endl;
    }
};

int main()
{
    HittingGame game;
    game.playGame();
    game.showResult();
    return 0;
}
```

OUTPUT:



C:\Users\AA\Desktop\lab 2 q4.exe

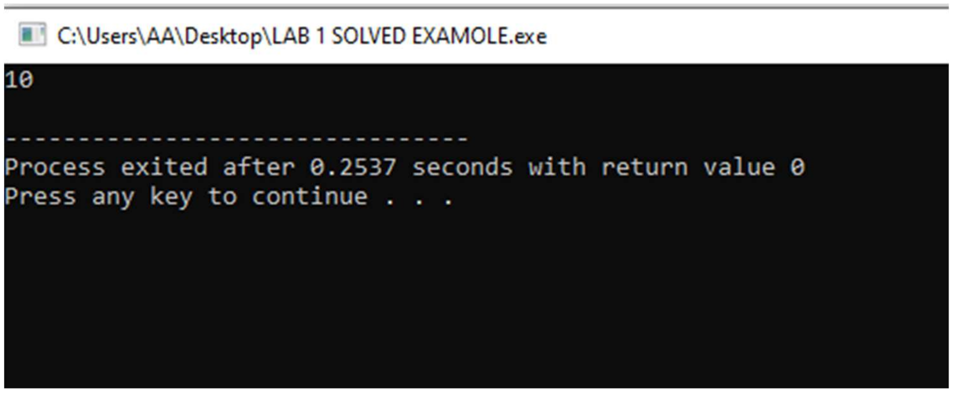
```
Number of players in your team: 7

Round 1
You got hit by the enemy team!

Round 2
Enemy got hit by your team!

Round 3
You got hit by the enemy team!

Round 4
Enemy got hit by your team!

Round 5
You got hit by the enemy team!

Round 6
You got hit by the enemy team!

Round 7
You got hit by the enemy team!

Your Team Hits: 2
Enemy Team Hits: 5
Enemy Team Wins!

--------------------------------
Process exited after 0.3271 seconds with return value 0
Press any key to continue . . .
```

SOLVED EXAMPLE:

SOURCE CODE:

```
#include <iostream>
using namespace std;
class firstprogram {
        private: // we declare a as private to hide it from outside
  int number1;
        public:
        void set(int input1){//set() function to set the value of a
          number1 = input1;
        }
        int get() { // get() function to return the value of a
  return number1;
        }
  };
// main function
int main() {
firstprogram myInstance;
        myInstance.set(10);
        cout << myInstance.get() << endl;
        return 0;
}
```

OUTPUT:



```
C:\Users\AA\Desktop\LAB 1 SOLVED EXAMOLE.exe
10

--------------------------------
Process exited after 0.2537 seconds with return value 0
Press any key to continue . . .
```