



COMSATS University Islamabad, Lahore Campus
Department of Computer Science

Sessional I – Semester Fall 2020

Course Title:	Object Oriented Programming-Lab			Course Code:	CSC241	Credit Hours:	4(3,1)
Course Instructor/s:	Ms. Muntaha Iqbal			Program Name:	BCE		
Semester:	3 rd	Batch:	FA19-BCE	Section:	A	Date:	29-10-2020
Time Allowed:	90 Minutes			Maximum Marks:	20		
Student's Name:	Hassaan Saleem			Reg. No.	FA19-BCE-024		

Important Instructions:

- Student is himself/herself responsible for successful submission of the exam on CU-Online.
- Your submission must include the following in a single pdf file.
 1. Code of all classes
 2. Snapshot of the output of submitted code.
- Don't waste your time in implementing irrelevant things, which are not asked.
- Copied exam will get zero credit.
- **Deadline: October 29, 2020 at 1:15 pm**

Question #	1	2	Total
Obtained Marks			
Total Marks	13	07	20

Task 1: CL05 -> PLO10 -> (A3, P2)

- Create a class named TempleRun_LevelOne with attributes player names and coin (initial value 10)
- Create static methods moveRight() -> coin -2 , moveLeft() -> coin -3, and jump() -> coin +5.
- Create another class named TempleRun_LevelTwo with a static method levelTwo(). This method will be called in TempleRunTest class only if coin value exceed 30.
- When levelTwo() method is called program should print the coin value and print a message "Congrats you are at level Two, Total".
- In TempleRunTest class call the above all methods. Note that you cannot call a method more than two times consecutively. Ask user for choice, if user press 1 call method moveRight(), if user press 2 call method moveLeft(), and if user press 3 call method jump. With every move, you have to print score.
- Player cannot move if coin value is zero. If coin value is zero or less it should print score with the message, "You can't move" and program should terminate.

- When levelTwo is achieved, program should print name of player, number of times player moved left, right and jumped.

Ans:

Code:

```
public class TempleRunTest
{

    public static void main(String[] args)
    {
        for(int i=1; i!=TempleRun_LevelOne.coin(); i++)
        {
            TempleRun_LevelOne.levelone();

            if (TempleRun_LevelOne.coin > 30) break;

        }

        if(TempleRun_LevelOne.coin() >30)
        {
            TempleRun_LevelTwo.leveltwo();
        }

        else
        {
            System.out.println("not sufficient score");
        }
    }
}
```

```
import java.util.Scanner;
```

```
public class TempleRun_LevelOne
{
    String playername;
    static int coin = 10;

    static void levelone()
    {
        String mr = "moveRight";

        String ml = "moveLeft";

        String jump = "jump";
```

```

Scanner s = new Scanner(System.in);

System.out.println("press 1 for move right press 2 for move left press 3 for jump");

s.nextLine();

String word1 = s.nextLine();

switch (word1)
{
    case "press1":
    {
        System.out.println(mr);
        coin = coin - 2;

        System.out.printf("the coin is %d \n",coin);
    }
    break;

    case "press2":
    {
        System.out.println(ml);
        coin = coin - 3;
        System.out.printf("the coin is %d \n",coin);
    }
    break;

    case "press3":
    {
        System.out.println(jump);
        coin = coin + 5;
        System.out.printf("the coin is %d \n",coin);
    }
    break;

    default:
    {
        System.out.println("in valid");
    }
    break;
}

static int coin()

{
    return coin;
}

}

```

```

public class TempleRun_LevelTwo
{
    static void leveltwo()
    {
        System.out.println("coin value is " + TempleRun_LevelOne.coin());

        System.out.println( "congratas you are now at level two" + "total is " + TempleRun_LevelOne.coin());
    }
}

```

Paste output:

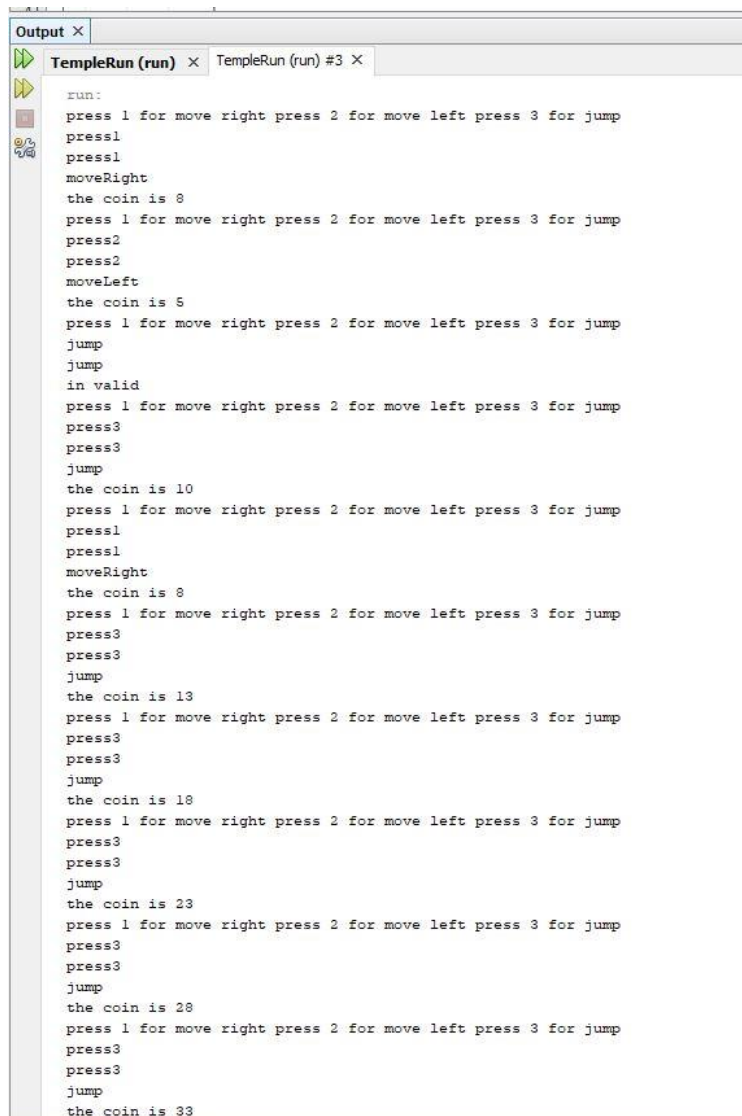
```

run:
press 1 for move right press 2 for move left press 3 for jump
press1
press1
moveRight
the coin is 8
press 1 for move right press 2 for move left press 3 for jump
press2
press2
moveLeft
the coin is 5
press 1 for move right press 2 for move left press 3 for jump
jump
jump
in valid
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 10
press 1 for move right press 2 for move left press 3 for jump
press1
press1
moveRight
the coin is 8
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 13
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 18

```

press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 23
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 28
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 33
coin value is 33
congratas you are now at level twototal is 33
BUILD SUCCESSFUL (total time: 1 minute 25 seconds)

Screenshot output:



```
Output X
TempleRun (run) x TempleRun (run) #3 x
run:
press 1 for move right press 2 for move left press 3 for jump
press1
press1
moveRight
the coin is 8
press 1 for move right press 2 for move left press 3 for jump
press2
press2
moveLeft
the coin is 5
press 1 for move right press 2 for move left press 3 for jump
jump
jump
in valid
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 10
press 1 for move right press 2 for move left press 3 for jump
press1
press1
moveRight
the coin is 8
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 13
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 18
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 23
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 28
press 1 for move right press 2 for move left press 3 for jump
press3
press3
jump
the coin is 33
```

Task 2:

- Create a class Employee with attributes employeeID, employeeName; monthlySalary (should be greater than 30000.00). monthlySalary is private attribute and would be exposed via getter and setter, while other two attributes are public.
- Create a constructor with employeeID attribute.
- Create another constructor with employeeID, and employeeName. In this constructor, call constructor with employeeID.
- Create a fully parametrized constructor. In this construct, call the constructor with employeeID, and employeeName.
- Create a method calculateYearlySalary(), which calculates and prints yearly salary of employee.

Carefully think about return type of method and instance Variables. No need to write test class.

Ans:

Code:

```
public class Employee
{
    double empolyeeid;
    String employeename;
    private double monthllysalary;

    public Employee(double empolyeeid)
    {
        this.empolyeeid = empolyeeid;
    }

    public Employee(double empolyeeid, String employeename)
    {
        this(empolyeeid);
        this.employeename = employeename;
    }

    public Employee(double empolyeeid, String employeename, double monthllysalary)
    {
        this(empolyeeid, employeename);

        setMonthllysalary(monthllysalary);
    }

    public void setMonthllysalary( double monthllysalary)
```

```

{
    if(monthlysalary > 30000.00)
    {
        this.monthlysalary = monthlysalary;
    }

    else
    {
        System.out.println("wrong input");
    }
}

public double getMonthlysalary()
{
    return monthlysalary;
}

public void calculateyearlysalary()
{
    double yearlysalary = monthlysalary * 12;

    System.out.println("the yearly salary is " + yearlysalary);
}

}

```

Paste output:

No output

Screenshot output:

No output

End of lab sessional.