Advanced Programming (CSE201), Quiz -1

Time allocated: 04:15pm – 4:35pm (20 minutes)

Instructions:

- This is a closed book quiz.
- Only reasonable and clearly mentioned assumptions (if any) would be accepted.
- For justifications, please be as concise as possible (2-3 sentences only)
- IIIT plagiarism policy is applicable if any such cases found
- Write your answers on a plain sheet that you can upload by taking a picture of the same (ensure low resolution so that the upload size is smaller)
- You can email the quiz solutions to **"ap-m2020-submission@iiitd.ac.in"**. Subject of the mail should be **<Rollno>_Quiz1**
- We will not consider any submission that is emailed beyond 4.45pm. It is your responsibility to ensure you email it on time. Please ensure you have proper internet connectivity as we are giving you sufficient extra time to send the email.

Question-1) "*US Open is one of the tennis tournaments. Roger and Nadal are tennis players that are participating in the US Open. Every tennis player can serve and volley. Every game in the tournament contributes to the ranking and the prize money that a player gets".*

Use object oriented programming principles to implement the above description in Java and identify the class relationships (if any). No need to code the main method.**Pseudocode implementation is sufficient as long as it can capture all OOP requirements**. **[5 marks]**

```
class TennisTournament {
  private TennisPlayer tennisPlayer1, tennisPlayer2;
  public TennisTournament(TennisPlayer tennisPlayer1, TennisPlayer tennisPlayer2) {
  this.tennisPlayer1 = tennisPlayer1;    //Association Relationship (+0.66 marks)
  this.tennisPlayer2 = tennisPlayer2;
  }
  public void playGame() {}
  public int computeRank() {}
  public int computePrizeMoney() {}
}

class TennisPlayer {
  public void serve() {}
  public void volley() {}
}
```

//The general Skeleton should be there even though it may not be exactly the same.

Line-1:	**[0.25 marks]**
Line-2:	**[0.25 marks]**
Line-3:	**[0.25 marks]**
Line-4:	**[0.5 marks]**
Line-5:	**[0.25 marks]**
Line-7:	**[0.25 marks]**
Line-8:	**[0.25 marks]**
Line-9:	**[0.25 marks]**
Line-11:	**[0.25 marks]**
Line-12:	**[0.25 marks]**
Line-13:	**[0.25 marks]**

Relationships:
USOpen is an instance of TennisTournament (+0.67 marks)
Roger and Nadal are instances of TennisPlayer (+0.67 marks)

Question-2) “*Apart from men’s singles games of US Open in which Roger and Nadal participate in, players also participate in other categories such as men’s double’s, women’s single, mixed doubles, etc. For players across all these categories, there would be changes in their ranking and prize money after every game, but the amount of prize money won would change depending on the category*”.

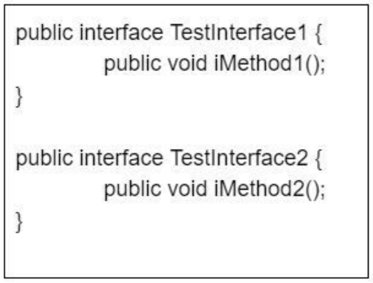
Use object oriented programming principles to implement the above description in Java and modify your code from the previous question. **Pseudocode implementation is sufficient as long as it can capture all OOP requirements.[2 marks]**

```
1. Interface TennisGameCategory {
2. computeRank();
3. computePrizeMoney();
4. }

5. class MensDoublesCategory implements TennisGameCategory {
   //must provide SOME DUMMY OR EMPTY implementation of methods declared in interface
   ...

6. }
```

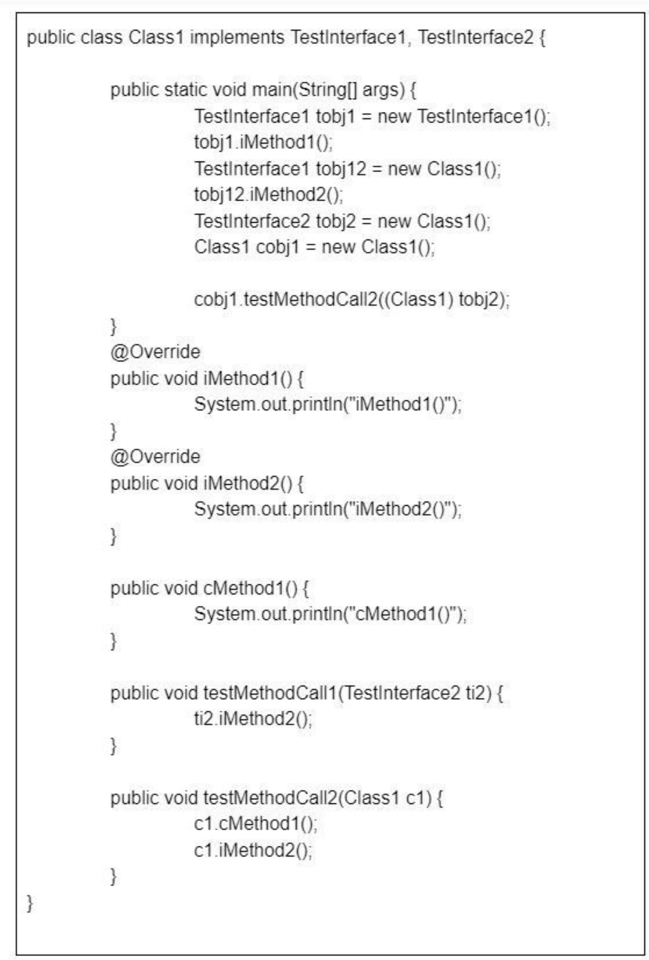
```
   7.  class MensSinglesCategory implements TennisGameCategory {
     //must provide SOME DUMMY OR EMPTY implementation of methods declared
in interface
       ...
   8.  }

   9.  class TennisTournament {
   10.  public void computePlayerRank(TennisGameCategory tgc) {
   11.  tgc.computeRank();
   12. }
   13. public void computePlayerPrizeMoney(TennisGameCategory tgc) {
   14.  tgc.computePrizeMoney();
 }
```

Line-1:	**[0.25 marks]**
Line-2&3:	**[0.25 marks]**
Line-5:	**[0.25 marks]**
Line-7:	**[0.25 marks]**
Line-10:	**[0.25 marks]**
Line-11:	**[0.25 marks]**
Line-13:	**[0.25 marks]**
Line-14:	**[0.25 marks]**

Question-3) Correct the program so that it works properly **[3 marks]**



```
public interface TestInterface1 {
        public void iMethod1();
}

public interface TestInterface2 {
        public void iMethod2();
}
```



```
public class Class1 implements TestInterface1, TestInterface2 {

        public static void main(String[] args) {
                TestInterface1 tobj1 = new TestInterface1();
                tobj1.iMethod1();
                TestInterface1 tobj12 = new Class1();
                tobj12.iMethod2();
                TestInterface2 tobj2 = new Class1();
                Class1 cobj1 = new Class1();

                cobj1.testMethodCall2((Class1) tobj2);
        }
        @Override
        public void iMethod1() {
                System.out.println("iMethod1()");
        }
        @Override
        public void iMethod2() {
                System.out.println("iMethod2()");
        }

        public void cMethod1() {
                System.out.println("cMethod1()");
        }

        public void testMethodCall1(TestInterface2 ti2) {
                ti2.iMethod2();
        }

        public void testMethodCall2(Class1 c1) {
                c1.cMethod1();
                c1.iMethod2();
        }
}
```

Class1 Errors:

1) Compiler error on instantiation of TestInterface1 -- Change to Class1 [+1.5 marks]
2) Compiler Error on invoking iMethod2() on TestInterface1 -- Change to TestInterface2 [+1.5 marks]