United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Mid Term Exam, Trimester: Fall 2022
Course Code: CSE-1115, Course Title: Object Oriented Programming
Total Marks: 30, Duration: 1 hour 45 minutes

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

<table>
<tr><td>1.</td><td>Write the output of the following program:

<table>
<tr>
<td>

```
public class GoT
{
  {
    System.out.println("Valar dohaeris");
  }
  public String name;
  public String house;
  public double impact;
  public float intent;
  public GoT()
  {
    System.out.println("Best TV series after Breaking
Bad");
  }
  public GoT(String name, String house)
  {
    this.name = name;
    this.house = house;
  }
  public GoT(double impact)
  {
    this("Daenerys","Targaryen");
    this.impact = impact;
  }
  public GoT(float intent)
  {
    this("Arya","Stark");
    this.intent = intent;
  }
```

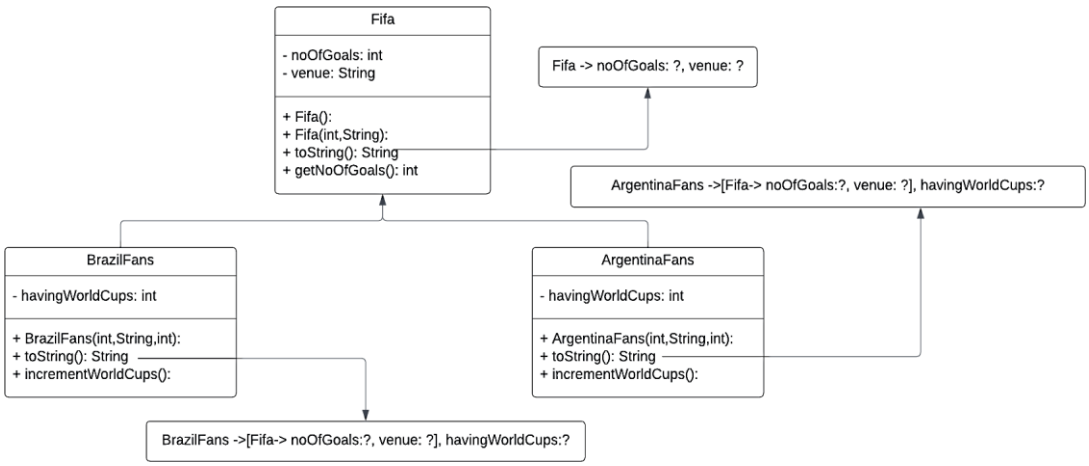
</td>
<td>

```
public void printFullName()
  {
    System.out.println(name + " " + house);
  }

public void printDetails()
  {
    printFullName();
    System.out.println("Impact: " + impact);
    System.out.println("Intent: " + intent);
  }
  {
    System.out.println("Valar morghulis");
  }
        public static void main(String[] args) {
          GoT ob1 = new GoT();
          ob1.name = "John Snow";
          ob1.house = "404";
          GoT ob2 = new GoT(4.8);
          ob1.printDetails();
          ob2.printDetails();
        }
}
```

</td>
</tr>
</table>
</td><td>6</td></tr>
<tr><td>2.</td><td>A Class named Movie containing following elements: name (String), origin (String), genre (String) and rating (Float). Among them, name cannot be accessed outside the class and origin can only be accessed in inherited class. Other two elements can be accessed from anywhere.

Write down the Movie Class which will contain following members:
a. Provide appropriate Access Modifier to the elements.
b. Create Necessary Methods to access name and origin from outside the class.
c. Two Parameterized Constructors where First Constructor will take all four elements and Second Constructor will take only name and genre.
d. A method named void Details() that will print the details of the Movie in following format:
<table>
<tr><td>You are watching SHUTTER ISLAND
Origin: USA
Genre: Thriller
Rating: 8.2</td></tr>
</table>
</td><td>6</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

3.	``` package transport; public class Vehicle { int noOfWheels; public Vehicle(int noOfWheels){ this.noOfWheels = noOfWheels; } } ``` ``` package publicTransport; public class Bus extends Vehicle { private int licenseNo; Bus(int licenseNo){ super.noOfWheels = 4; this.licenseNo = licenseNo; } } ``` Rewrite the above blocks of code with the following changes. • Make sure no other classes have direct access to 'noOfWheels' in 'Vehicle'. • Make Sure 'noOfWheels' in 'Vehicle' is read-only both from inside and outside of the 'transport' package. (the value can be set only within the constructor) • Make sure the '**`Bus extends Vehicle`**' line does not throw any error. • Resolve any error that might be thrown by the constructor of class '**`Bus`**'. • Enable read and write access to 'licenseNo' of the class 'Bus' without changing the given access modifier.	**6**
4	All of you know that FIFA 2022 with Brazil, Argentina controversy is a trendy topic. Given the following UML diagram, which has three classes: Fifa, BrazilFans, and ArgentinaFans. Although each detail is depicted in the UML diagram, some of the more complex parts are written here for convenience.  There are 3 Classes. **Fifa, BrazilFans and ArgentinaFans.** **Fifa()** constructor prints **" Who will be winner?"** message. The first line of the **Fifa(int,String)** constructor body is called **Fifa()** constructor with the help of **this** keyword, and the remaining two lines instantiate variables using this keyword. **toString()** methods return a string with the given format.	6

BrazilFans is a derived class of **Fifa**, which has a constructor with three parameters, the first two of which are passed to the base class, and the last one is used to instantiate variables using **this** keyword. **toString()** methods return a string with the given format. The **incrementWorldCups()** methods increase the value of **havingWorldCups** by 1 when it is invoked.

ArgentinaFans is a derived class of **Fifa**, which has a constructor with three parameters, the first two of which are passed to the base class, and the last one is used to instantiate variables using **this** keyword. **toString()** methods return a string with the given format. The **incrementWorldCups()** methods increase the value of **havingWorldCups** by 1 when it is invoked.
Mid is another class that contains **main()** methods. In the **main()** method, after creating objects of **ArgentinaFans** and **BrazilFans**, if the **noOfGoals** of Argentina is greater than Brazil, then the **incrementWorldCups()** method of **ArgentinaFans** is called; otherwise, the **incrementWorldCups()** method of **BrazilFans** is called. Write the whole program appropriately so that when it executes, you get the following output:

```
Who will be winner?
Argentina will win
Who will be winner?
Brazil will win
ArgentinaFans -> [Fifa -> noofGoals: 10, venue:Qatar], havingWorldCups:2
BrazilFans -> [Fifa -> noofGoals: 7, venue:Qatar], havingWorldCups:5
ArgentinaFans -> [Fifa -> noofGoals: 10, venue:Qatar], havingWorldCups:3
BrazilFans -> [Fifa -> noofGoals: 7, venue:Qatar], havingWorldCups:5
```

5

```java
class PClass{
    void mFnc(){
        System.out.println("Hello from P Class!");
    }

    void mFnc(double d1){
        System.out.println("Double value: " + d1);
    }
}
```

```java
class CClass extends PClass{
    void mFnc(){
        System.out.println("Hello from C Class!");
        super.mFnc(11.22);
    }

    void mFnc(int a2, double d2){
        mFnc(d2);
        System.out.println("Integer value: " + a2);
    }
}
```

```java
class Main{
    public static void main(String[] args) {
        PClass pObj = new PClass();
        CClass cObj = new CClass();

        pObj.mFnc();
        cObj.mFnc();
        cObj.mFnc(10, 2.99);
        cObj.mFnc(3.145);
    }
}
```

Check the attached Code Snippet and answer following questions:

1. What is the Output of the Main Class?
2. "In the "**CClass"** which is a subclass of **"PClass",** both ***Method Overriding*** and ***Method Overloading*** has been demonstrated"- Explain the statement using the examples from the code snippet.

6