United International **University** (**UIU**)

Dept. of Computer Science and Engineering (CSE)

Midterm Assessment Year: **2022** Semester: **Summer**
Course: **CSE 1115** Title: **Object Oriented Programming** Section: **(A-F)**
Marks: **30** Time: **1 Hour 45 minutes**

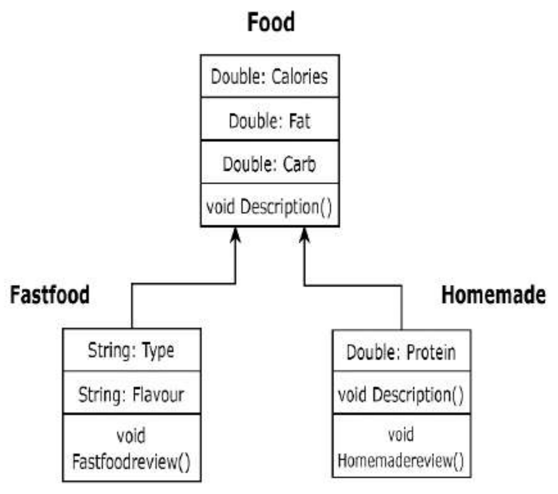
[Any examinee found adopting unfair means will be expelled from the trimester/program as per UIU disciplinary rules.]

There are **5 (Five)** questions. Answer **all 5 (Five)** questions. All questions are of values indicated on the right-hand margin.

Q1. Write a Java program for the followings: **[6]**

i. Create a package named americanSuperhero.
ii. In americanSuperhero, create two classes named SpiderMan.class and SuperMan.class.
iii. In SpiderMan class, declare two instance variables of String type movieName and directedBy, and initialize these variables in constructor with this reference keyword.
iv. Write a method named show() to print the instance variables in SpiderMan class.
v. Now from the main method of the SuperMan class create an object of SpiderMan class with movieName = “The Amazing Spider-Man 2” and directedBy = “Marc Webb”, and call the show() method of SpiderMan class.

Q2. Consider the following diagram provided below. The Food class is given. **[6]**



```
public class Food
{
   Double calories, fat, carb;
   Food(Double calories,  Double fat,
          Double carb )
   {
      this.calories=calories;
      this.fat=fat;
      this.carb=carb;
   }
   void description()
   {
      System.out.println("Inside Food class");
   }
}
```

Tasks:

i. Create two subclasses of Food class named Fastfood and Homemade as shown in the diagram.
ii. In the Fastfood class, create a five parameterized constructor of this class. You must use super(parameterized) constructor to access the superclass attributes.
iii. Create a method named Fastfoodreview() in the Fastfood class. The method should print “Inside the review method of Fastfood class.”
iv. In the Homemade class, create a four parameterized constructor of this class. You must use super(parameterized) constructor to access the superclass attributes.
v. Create a method named Homemadereview() in the Homemade class. The method should print “Inside the review method of Homemade class.”
vi. Override the Description() method of Food class in the Homemade class.

vii. Create a main class named Foodmain, inside that main class perform subclass polymorphism to invoke the Homemadereview() method through a Food reference variable f1, and invoke the Fastfoodreview() method by a Food reference variable f2.

Q3. a. Consider the following code blocks and write only the output from the main method with the sequence as they will appear in the console: **[3]**

```
class Mid {
    int x = 10;
    {
        x = 20;
        System.out.println("@Block=" + x);
    }
    Mid(int x1, int x2) {
        x = x1 + x2;
        System.out.println("@Constructor-2=" + x);
    }
    Mid(int x1) {
        this(100, 200);
        x = x1;
        System.out.println("@Constructor-1=" + x);
    }
    Mid() {
        this(50);
        x = 30;
        System.out.println("@Constructor-0=" + x);
    }}
```

```
public class InitBlock {

    public static void main(String[] args) {

        Mid obj1 = new Mid(30);

        System.out.println("@End=" + obj1.x);

    }

}
```

Q3. b. Consider the following code block: **[3]**

```
class Point {
    int x;
    int y;

    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
public class PointDemo {
    void resetPoint1(Point p) {
        p = new Point(0, 0);
    }
    void resetPoint2(Point p) {
        p.x = 0;
        p.y = 0;
    }
    public static void main(String[] args) {
        Point p1 = new Point(3, 5);
        Point p2 = new Point(10, 15);

        PointDemo demo = new PointDemo();
        demo.resetPoint1(p1);
        System.out.println("(" + p1.x + ", " + p1.y + ")"); // Question (i)
25          demo.resetPoint2(p2);
26          System.out.println("(" + p2.x + ", " + p2.y + ")"); // Question (i)
27
28          p2 = new Point(9, 9); //  Question (ii)
29      }
30  }
```

i. What will be the outputs from Line No: 24 & 26?

ii. Explain the consequences of Line No: 28 in terms of Garbage Collection

Q4. There is a class Person having member variables id, name, height (meter in double), weight (kg in double): **[6]**

```
class Person{
        double id, height, weight;
        String name;
}
```

You can assume that the constructor needed for the code is already written in the Person class. There is another class Test, and the class looks like this:

```
public class Test{
        public static void main(String args[]){
                Person p = new Person();
                p.id = 1; //error
                p.name = “Steven”; //error
}
}
```

When you try to access the id and name of Person p directly, it throws an error (see the Test class code).

Tasks:

i. What is the reason for this error? Modify the Person class without changing the access modifier so that we can access the id and name from the Test class.

ii. Suppose you want to measure BMI for a Person “p1” from another class in the same file. Formula to calculate BMI is:

BMI = weight / (height * height)

Now modify the Person class to measure BMI where height and weight is no way accessible from the other classes than the Person class itself.

iii. Declare an array of length 3 where each index in the array refers to a Person object.

Q5. a. Write the output of the following code: **[3]**

```
public class Comics{
   public void foo(){
      System.out.println("foo");
   }
   public void bar(){
      System.out.println("bar");
   }
}
```

```
public class Main {
   public static void main(String[]
      args) {Marvel obj = new
      Marvel();
      obj.foo();
      obj.foo(50);
      obj.fubar();
   }
}
```

```
public class Marvel extends Comics{
   public
      Marvel(){
      this(1000);
   }
   public Marvel(int val){
      System.out.println("Value: " + val);
   }
   public void foo(){
      super.foo();
      System.out.println("Tony Stark");
   }
   public void foo(double val){
      System.out.println("Steve Rogers");
      System.out.println("Value: " + val);
   }
   public void fubar(){
      bar();
   }}
```

Q5. b. Write the output of the following code: **[3]**

```
public abstract class Shape {
      public abstract void printArea();
}
```

```
public class Rectangle extends Shape {
      double width, height;
      public Rectangle(double width, double
      height)
      {
            this.width = width;
            this.height = height;
      }
      public void printArea(){
            double area = width * height;
            System.out.println("Area: " + area);
      }}
```

```
public class Square extends Shape {
double side;
public Rectangle(double side) {
side = side;
}

public void printArea(){
double area = side * side;
System.out.println(“Area” + area);
 }
}
```

```
public class Main {
   public static void main(String[] a) {

         Shape sh1, sh2;
         Rectangle r1;
         Square s1;

         sh1 = new Square(10);
         sh2 = new Rectangle(10, 20);
         s1 = new Square(5);
         R1 = new Rectangle(5, 10);

         r1.printArea();
         s1.printArea();
         sh1.printArea();


         sh2.printArea();


      }
}
```