United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Exam, Trimester: Fall 2024
Course Code: CSE-1115, Course Title: Object Oriented Programming
Total Marks: 30, Duration: 1 Hour 30 Minutes

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

QUESTION 1   [2+2+2 MARKS]

Consider the following codes:

```
public class Employee{
 public String name;
 private int employee_id;
 private double salary;

// Write constructor to initialize name, employee
// id and salary with this reference keyword;

 // Write getter method for salary;
 // Write setter method for salary;
```

```
public static void main(String[] args) {
     Employee e1 = new Employee("Prof. Albert Einstein",
                 19, 2000.50 );
     System.out.println(e1.name);
  }
}
```

- Write constructor to initialize name, employee_id and salary with this reference keyword.
- Write getter method for salary variable.
- Write setter method for salary variable.

QUESTION 2   [3+3 MARKS]

Consider the following codes:

```
class Vehicle {
 protected String brand;
 protected int serial;

 public Vehicle(String brand) {
    this.brand = brand;
 }

 public Vehicle(int serial) {
    this.serial = serial;
 }

 public Vehicle(String brand. int serial){
    this(brand);
    this(serial);
 }
```

```
public boolean honk() {
     System.out.println("Tuut, tuut!");
     return true;
 }
}

public class Car extends Vehicle {
 private String modelName;

 // Invoke parent class constructor;
 // Invoking overriding method;

 public static void main(String[] args) {
    Car car1= new Car("Honda", 20245 , "Integra");
    car1.honk();
    System.out.println(car1.brand + " " +car1.modelName);
  }
}
```

1. Only one constructor call (this(…) or super(…)) is allowed, and it must be the first statement.

- Write the constructor of car class and invoke the parent class constructor.

- Override honk() method and print "A vehicle horn is a sound-making device." then invoke the overridden method.

QUESTION 3 [6 MARKS]

Write output of the following codes:

```
class Operations {
  static int total = 20;

  static {
    System.out.println("Global Tech Academy");
  }

  static int square(int y) {
    return y * y;
  }

  public static void Increment() {
    total++;
    System.out.println(total);
  }
```

```
public static void main(String args[]) {

    Operations.Increment();

    Operations obj1 = new Operations();

    obj1.Increment();

    Operations obj2 = new Operations();

    obj2.Increment();

    System.out.println(obj1.total);

    int result = Operations.square(Operations.total);

    System.out.println(result);

  }
}
```

QUESTION 4 [6 MARKS]

Write output of the following codes:

```
public class Sum {
 int x, y, z;

 {
  System.out.println(x+y+z);
 }

 {
  x = 11;
  y = 22;
  z = 33;
  System.out.println(x+y+z);
 }

public Sum(int x) {
  this.x = x;
  System.out.println(x+y+z);
 }
public Sum(int x, int y) {
  this(500);
  this.x = x;
  this.y = y;
  System.out.println(x+y+z);
 }

public Sum(int x, int y, int z) {
  this(50, 20);
  this.x = x;
  this.y = y;
  this.z = z;
  System.out.println(x+y+z);
 }

public void display(){
  this.display("The pen is stronger than the sword.");
 }

public void display(String str){
  System.out.println(str);
 }

public static void main(String args[])
 {
  Sum s = new Sum(100, 150, 300);
  s.display();
 }
}
```

QUESTION 5 [6 MARKS]

class Figure {
 	
  void render() {
 	
    System.out.println("rendering...");
 	
  }
 	
}
	class Oval extends Figure {
	
	  void render() {
	
	    System.out.println("rendering oval...");
	
	  }
	
	}

class Square extends Figure {
 	
  void render() {
 	
    System.out.println("rendering square...");
 	
  }
 	
}
	class Polygon extends Figure {
	
	  void render() {
	
	    System.out.println("rendering polygon...");
	
	  }
	
	}

Write a class named Demo, which contains the main() method. In the main() method, create a heterogeneous array named f of the Figure class, where the array length is three. Index f[0] should be created with the Square class, f[1] with the Oval class, and f[2] with the Polygon class. Finally, iterate through a loop to call the render() method of each object.

```
Figure[] f = new Figure[3

f[0] = new Square()
f[1] = new Oval();
f[2] = new Polygon

for(Figure t: f){
    t.render();
}
```