United International University
CSI 211: Object Oriented Programming, Spring 2018
Mid Term Exam
Total Marks: 30, Time: 1 hour 45 minutes

Answer all 5 questions (5 X 6 = 30).

Question 1

(a) Complete the **Student** class below by adding appropriate attributes and/or methods such that running the main method provides the following expected output. You must not modify the main method. [4]

```
public class Student {
	private String name, id;
	private double cgpa;
	public static void main(String[] args){
		Student s1 = new Student("011162101", "Kratos", 3.0);
		Student s2 = new Student("011162102", "Thanos", 4.0);

		System.out.println("Before swap");
		System.out.println(s1);
		System.out.println(s2);

		swap(s1, s2);
		System.out.println("After swap");
		System.out.println(s1);
		System.out.println(s2);
	}
}
```

Expected Output:

```
Before swap
011162101 Kratos 3.0
011162102 Thanos 4.0
After swap
011162102 Thanos 4.0
011162101 Kratos 3.0
```

Or

Write three classes: ***Animal***, ***Cat***, and ***AnimalHeredityTest***. The ***Animal*** class has three **private** variables: ***vegetarian***, ***eats***, and ***noOfLegs***. On the other hand, ***Cat*** class has only one private variable: ***color***. ***Cat*** class inherits ***Animal*** Class. In ***AnimalHeredityTest*** class, create an object ***c1*** of ***Cat*** class and display the properties of the object ***c1*** *and* ***its parent's***. Use any of the Object Oriented feature to access the private variables of the base and child classes. Note: You need to write code for all 3 classes. [4]

(b) What is probably wrong with the following code? Explain briefly. [2]

```
package p1;

public class Test3 {
	int addTwoNumbers(int a, int b) {
		return a + b;
	}
}
```

```
package p2;
import p1.*;
public class Test1 {
  public static void main(String args[]){
	Test3 obj = new Test3();
	obj.addTwoNumbers(10, 21);
  }
}
```

Question 2

(a) What will be the output of the following code? Answer one of them. [2]

```
public class E {
 int a;
 public int getA() { return a;     }
 public void setA(int a) {  a = a; }
 E show() {  return this;  }
 public static void main(String[] args) {
       E obj = new E();
       obj.setA(10);
       System.out.println(obj.getA());
       E obj2 = obj.show();
       System.out.println(obj2.getA());
 }
}
```

OR

```
public class E {
public static void main(String[] args){
		int a = 10;
		double b = 5;
		String c = "Bazinga";
		System.out.println(a + b + c);
		System.out.println(c + a + b);
		System.out.println(b + a + c);
	}
}
```

(b) Write down the output of the program below. [4]

```
public class TestAnimal {
  public static void main(String[] args){
        Animal a = new Animal("Rat");
        Dinosaur d = new TRex();
        a.display();
        d.display();

        a=d;
        a.display();
        d =(Dinosaur)a;
        d.display();
  }
}

class Animal{
  String name;
  float weight;
  Animal(String n){ name = n;  }

  public void display(){
   System.out.println("Animal can be Omnivorous.");
  }
}
```

```
class Dinosaur extends Animal{
 Dinosaur(String n) {
   super(n);
 }

 public void display(){
   System.out.println("Dinosaur mostly Herbivorous");
 }
}

class TRex extends Dinosaur{

 TRex(){
   super("TRex");
 }

 public void display(){
   super.display();
   System.out.println("but "+name+" is carnivorous");
 }
}
```

Question 3

(a) "The package is both a naming and a visibility control mechanism."-explain the statement. [2]

(b) Briefly explain with an appropriate example the differences between Class, Reference and Object. [2]

(c) Can an **abstract method** be declared **private**? Explain your answer. [2]

Question 4

(a) Create two concrete Java classes **Cat** and **Dog** such that the main method produces the expected output. You must not modify the **Pet** and **Main** class and it **is not necessary to re-write** them in your answer script. Note that a Cat's body gets heated if its body temperature is at least 50 units whereas for a dog it is 80 units. [4]

```
abstract public class Pet {
    int bodyTemperature;

    public Pet(int bodyTemperature) {
        this.bodyTemperature = bodyTemperature;
    }

    abstract void make_noise();
    boolean is_heated() {
        return (bodyTemperature >= 80);
    }
    final void sleep() {
        System.out.println("Pet sleeping");
    }
}
```

```
public class Main {

    public static void main(String[] args) {
        Pet [] pets = new Pet[2];
        pets[0] = new Cat(50);
        pets[1] = new Dog(60);

        for (Pet p : pets) {
            p.make_noise();
            System.out.println("Warm: " + p.is_heated());
            p.sleep();
        }
    }
}
```

Expected output:

```
Meow
Warm: true
Pet sleeping
Ghew
Warm: false
Pet sleeping
```

(b) Assume there is a java class named ***Voter*** which has 3 attributes ***name, voterId*** and ***age***. Carefully observe the code below; the **output** of the code is **false** as v1 and v2 are 2 different objects with 2 different names. But in reality v1 and v2 refer to the same person; v1 is using his first name and v2 using last name.

What changes do you need to implement in Voter class so that the output of following code segment is **true**? You do not need to write the complete code of ***Voter*** class, write just the code segment that is required for the expected output. [2]

```
public static void main(String[] args) {
        Voter v1 = new Voter("Hasan", "001227593", 70);
        Voter v2 = new Voter("Mahmud", "001227593", 70);
        boolean isSamePerson = v1.equals(v2);
        System.out.println(isSamePerson);
}
```

Question 5

(a) Consider the Vector2D class. [3]

We want to add ***multiplication*** methods in this class. If a vector is **multiplied by a real number**, then both ***x*** and ***y*** values are simply multiplied by that real number. If it is **multiplied by another vector *v***, then it will follow the rule of dot product. **Dot product of two vectors u & v , u.v = u.x*v.x + u.y*v.y.**

In this Vector2D class, write two overloaded methods for ***multiplication***; 1) **one method takes a double parameter,** does the real number multiplication, then returns a Vector2D object and 2) the **other one takes a Vector2D parameter**, does the dot product and then return a double number.

```
public class Vector2D{
    double x, y;
    Vector2D(){
        x = 0;
        y = 0;
    }
    Vector2D(double x, double y){
        this.x = x;
        this.y = y;
    }
}
```

(b) Find and fix the errors in the given code snippet. For each error, point out the error, explain why it is an error and write down a possible way to fix it. You can edit any line of code but you are not allowed to delete any line of code. [3]

```
class A{
    public static final int var;
    public int par;
    static{
        var = 10;
        par = 5;
        System.out.println("Static block in A");
    }
    void meth(){
        var = 15;
        System.out.println("Method in class A");
    }
}
class B extends A{
    final void meth(){
        var += 20;
        System.out.println("Method in class B");
    }
}
```

```
final class C extends B{
    int star;
    void meth(){
        System.out.println("Method in class C");
    }

    public static void main(String[] args) {
        B obj1 = new B();
        B obj2 = new B();
        obj1.meth();
        star = 100;
        System.out.println("par = " + obj1.par + ",
var = "+ obj2.var + ", star = " + star);
    }
}
```