United International University (UIU)

Dept. of Computer Science and Engineering (CSE)

MID TERM EXAM :: Spring 2019

Course Code: **CSI 211** Course Title: **Object-Oriented Programming**

Date: **7/3/19** Total Marks: **30** Time: **1 Hour 45 mins**

1. Write the output of the following code: [5]

```
public class Zoo {
    public static void main(String[] args) {
        Animal a = new Animal();
        Bird b = new Magpie();
        Magpie c = (Magpie) b;

        a.fly();
        b.fly();
        ((Magpie)b).fly(15);
        c.fly();

        a.eat();
        b.eat();
        c.eat();
    }
}
```

```
class Magpie extends Bird{
    Magpie(){
        System.out.println("A magpie is created.");
    }
    void fly(){
        System.out.println("Magpie is flying...");
    }
    void fly(int speed){
        System.out.println( "Magpie is flying at speed: "
            + speed);
    }
    void eat(){
        System.out.println( "Magpie is eating.");
    }
}
```

```
class Bird extends Animal{
    Bird(){
        System.out.println("A bird is created.");
    }
    void fly(){
        System.out.println("Flying...");
    }
}
```

```
class Animal{
    Animal() {
        System.out.println( "An animal is created.");
    }
    void fly(){
        System.out.println("Don't know if I can fly!");
    }
    void eat(){
        System.out.println("Eating...");
    }
}
```

2. Consider the following **FruitShop** class. Now, write a class **Fruit** in such a way that **FruitShop** class will give expected output as shown below. [5]

Class FruitShop

```
public class FruitShop {
    public static void main(String[] args) {
        Fruit fruit1 = new Fruit("Apple", 3.5, 110);
        Fruit fruit2 = new Fruit("Mango", 5, 90);

        fruit1.reducePricePerKG(10);
        fruit2.increasePricePerKG(20);

        fruit1.printDetails();
        fruit2.printDetails();
    }
}
```

Output

```
Fruit Details:
Name: Apple
Weight: 3.5kg
Price per kg: 100.0
Total price: 350.0

Fruit Details:
Name: Mango
Weight: 5.0kg
Price per kg: 110.0
Total price: 550.0
```

3 a. Consider the following classes. Will the code run without errors? If so, please specify the errors. Otherwise, write the output of the code. [4]

```
class B{
    void a(){
        System.out.println("I am first A inside B");
    }
}

class C extends B{
    void a(int a){
        System.out.println("I am second a inside C, my
val is " + a);
    }
}
```

```
public class D{
    public static void main(String[] args) {
        B obj = new C();
        obj.a();
        obj.a(5);
    }
}
```

b. Is it a must for abstract classes to have abstract method/methods? [1]

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

```
public class Course {
    String name, id;
    int credit;
    public Course(String name, String id, int credit) {
     this.name = name;
     this.id = id;
     this.credit = credit;
    }
    public void display() {
     System.out.printf("%s-%s-%d\n", name, id, credit);
    }
    public void updateCourse(Course c1){
     this.name = c1.name;
     this.id = c1.id;
     c1.credit--;
    }
}
```

```
public class Application {
    public static void main(String[] args)  {
    Course c1=new Course("OOP","CSI211",3);
    Course c2=new Course("SPL","CSI121",1);
     c1.display();
     c2.display();
     c1.updateCourse(c2);
     c1.display();
     c2.display();
    }
}
```

b. Fix the following code. You cannot remove any lines. You can only add or edit existing lines. [2]

```
package zoo;

class Animal {
    String name;
    protected int legs;

    Animal(String name, int legs){
        this.name = name;
        this.legs = legs;
    }

    void print(){
        System.out.println(name + " " + legs);
    }
}
```

```
package zoo.birds;

class Bird extends Animal{
    String color;

    Bird(){ }

    Bird(String name, int legs, String color){
        this(name, legs);
        super.color = color;
    }

    void print(){
        System.out.println(name + " " +
            legs + " " + color);
    }
}
```

5. a. Will the code run without errors? If so, please specify the errors. Otherwise, write the output of the following code. [2.5]

```
class CrazyVars{

    static int i;
    int j;

    void incrementI(){
        CrazyVars.i++;
    }

    void incrementJ(){
        this.j++;
    }
}
```

```
public class Execution{

    public static void main(String[] args){
        CrazyVars workerOne = new CrazyVars();
        CrazyVars workerTwo = new CrazyVars();
        workerOne.incrementI();
        workerTwo.incrementJ();
        workerTwo.incrementI();
        workerOne.incrementJ();
        System.out.println(workerOne.i + workerTwo.j);
    }
}
```

b. Explain what is wrong with this code. Fix the code. You **CANNOT** add/delete any line. [2.5]

```
interface ABC {
                String s = "123";
                void method1(int a);
                void greeter();
 }
```

```
class C implements ABC{
  String s = "new";
  public void greeter() {
    System.out.println("Hi");
  }
}
```

6. a. Write the output of the following code: [2.5]

```
class SwapFunc {

    static void swapReference(RefFunc a, RefFunc b) {
        RefFunc c = a;
        a = b;
        b = c;

    }

}
```

```
public class RefFunc {
    int a;
    int b;
    public RefFunc(int a, int b) {
        this.a = a;
        this.b = b;
    }
    public static void main(String[] args) {
        RefFunc obj1 = new RefFunc(5, 6);
        RefFunc obj2 = new RefFunc(7, 8);
        SwapFunc.swapReference(obj1, obj2);
        System.out.println(obj1.a + "\t" + obj2.b);
    }
}
```

b. Will the code run without errors? If so, please specify the errors. Otherwise, write the output of the following code. [2.5]

```
abstract class Bike {
    int brandType;
    private abstract void velocity();
    abstract void showType() {
        System.out.println("brandType is " +
brandType);
    }
    public void firstCall() {
        System.out.println("A bike is being made");
    }
}
```

```
public class Runner extends Bike {
    protected void firstCall() {
        super.firstCall();
        System.out.println("A Runner is being made");
    }
    private void velocity() {
        System.out.println("Max velocity is 125");
    }
    public static void main(String[] args) {
        Bike obj = new Runner();
        obj.firstCall();

    }
}
```