United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Exam **:: Trimester: Fall - 2019**

Course Code: CSE 1115 Course Title: Object Oriented Programming

Total Marks: **30** Duration: **1 hour 45 minutes**

Question 1 [7 Marks]

a) Create a java class named **Calculator**. Calculator class has 3 overloaded **add** methods: **[3]**
1) The first **add** method takes **2 integers** as **parameter** and **returns** their sum.
2) The second **add** method takes **1 integer** and **1 double** as **parameter** and **returns** their sum.
3) The third **add** method takes **1 integer** and **2 doubles** as **parameter** and **returns** their sum.

After creating the class, create an object of **Calculator** and **call all three overloaded add methods** using appropriate parameters.

b) Write an abstract class named **Car** which has 3 protected instance variables: [(1) **model** (String), (2) **color** (String), (3) **maxSpeed** (int)] and **two abstract** instance methods: [(1) **accelerate()** and (2) **brake()**] **[2]**

Create another classes **Audi** by inheriting the **Car** class. You should provide necessary method(s) and a constructor that initializes all the instance variables in **Audi** class. **[2]**

Question 2 [6 Marks]

a) Carefully consider the following program. **[1+3]**

```java
class BankAccount{
    private String name;
    private double balance;

    BankAccount(String _name,
        double initialDeposit){
        name = _name;
        balance = initialDeposit;
    }
    void deposit(double amount){
        balance = balance + amount;
    }
    void withdraw(double amount){
        if(amount <= balance){
            balance = balance - amount;
        }
    }
    double getBalance(){
        return balance;
    }
    String getName(){
        return name;
    }
}
```

```java
class Test {
    public static void main(String[] args) {
        BankAccount gatesAccount;                                  // 1
        gatesAccount = new BankAccount("Bill Gates", 500.0);       // 2
        BankAccount jobsAccount = new BankAccount("Steve Jobs", 0.0); // 3
        gatesAccount.withdraw(150.0);
        jobsAccount.deposit(400.0);

        System.out.println(gatesAccount.getName() + " " + gatesAccount.getBalance());
        System.out.println(jobsAccount.getName() + " " + jobsAccount.getBalance());
        jobsAccount = null;                                        // 4
    }
}
```

i. What is the output of the program?
ii. Draw diagrams of all the objects and references after each of the **lines with a comment at the end** inside main.

b) What is the output of the following program? **[2]**

```
class Product{
    String productId;
    double price;
    Product(String id, double _price){
        productId = id;
        price = _price;
    }
}
```

```
class Modifier{
    void updateId(Product p,
       String newId){
        p.productId = newId;
    }
    void updatePrice(double curPrice,
       double newPrice){
        curPrice = newPrice;
    }
    void updateIdV2(Product p,
       String newId){
        p = new Product(newId, p.price);
    }
    void updatePriceV2(Product p,
      double newPrice){
        p.price = newPrice;
    }
}
```

```
public class Test {
    public static void main(String[] args) {
        Modifier m = new Modifier();
        Product p= new Product("L-124", 450.0);

        m.updateId(p, "L-567");
        System.out.println(p.productId);
        m.updatePrice(p.price, 470.0);
        System.out.println(p.price);

        m.updateIdV2(p, "L-135");
        System.out.println(p.productId);
        m.updatePriceV2(p, 490.0);
        System.out.println(p.price);
    }
}
```

Question 3 [6 Marks]

a) Consider the following code carefully: **[3+1]**

```
class Book {
    String name;
    String author;
    Book(String n, String a){
        name=n;
        author=a;
        System.out.println("This is a book");
    }
}
```

```
public class TextBook extends Book{
   String edition;
   TextBook(){
     System.out.println("This is a textbook");
   }
   public void setGenre(String e){
     edition=e;
   }
}
```

```
public class Novel extends Book{
   Novel(String a, String b){
     System.out.println(name+" is a novel");
   }
}
```

```
public class Test {
    public static void main(String[] args){
        Novel N = new Novel("Himu Mama",
                           "Humayun Ahmed");
    }
}
```

i. Write down the correct version of the code. You are **not allowed** to **remove** any line of the code. You can only **modify** the given lines or **add new lines**.

ii. Write the output of the **corrected** code.

b) Is the following code snippet allowed? Give your answer as 'yes' or 'no' and provide proper reason behind your answer. **[2]**

```
class Vertebrate{
   public final void get(){
    System.out.println("This is a vertebrate");
   }
}
```

```
class Human extends Vertebrate{
  @override
  public void get(){
    System.out.println("This is a human");
  }
}
```

OR

Consider the following Main class and the Output section. You have to write a class named "Fraction" to generate the desired output. **[2]**

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

        Fraction f1 = new Fraction();
        System.out.println(f1.num + "/" + f1.denom);
        Fraction f2 = new Fraction(2, 5);
        System.out.println(f2.num + "/" + f2.denom);
    }
}
```

Expected Output:

```
1/2
2/5
```

Question 4 [6 Marks]

a) The following code snippet contains four java classes. 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 are only allowed to add or modify any existing line of code**. But you **cannot** change **package name**, **cannot modify access modifier of the instance variables of any class** and **cannot delete any existing line**. **You are allowed to modify access modifier of any method.** **[4]**

```
package pack1;

public class Vehicle {
   protected int id;
   String name;
   public Vehicle(String name){
       this.name=name;
   }
}
```

```
package pack1.pack2;
public class Car extends Vehicle {
  public Car(String name, int id){
      super(name);
      this.id=id;
  }
  void print(){
       System.out.println(name + " " + id);
   }
}
```

```
package pack1;

public class Bike extends Vehicle {

   public Bike(String name, int id){
       super(name);
       this.id=id;
   }
   void print(){
       System.out.println(name+ " " + id);
   }
}
```

```
package pack1.pack2;
import pack1.*;

public class Main {
   public static void  main(String args[]){
       Vehicle v = new Vehicle("Cart");
       System.out.println( v.name);
       Car c=new Car("BMW", 101);
       c.print();
       Bike b=new Bike("Pulsar", 102);
       b.print();
   }
}
```

b) Observe the following code. Show how many objects are eligible for garbage collection after execute the following code. (Show using proper diagrams.) **[2]**

```
class Player{
   String name;
   String position;

   Player(String a,
     String b){
       name = a;
       position=b;
   }
}
```

```
public class Main {
   static void changePosition(Player p, String newPosition){
       p.position = newPosition;
   }
   public static void main(String[] args) {
       Player p1 = new Player("Ramos","Defence");
       Player p2 = new Player("Modric", "Midfield");
       Player p3 = new Player("Neymar", "LeftWing");
       changePosition(new Player("Messi", "Striker"), "MidField");
       p2 = p3;
   }
}
```

Question 5 [5 Marks]

a) Observe the following code carefully and write the **output(s)**. [3]

```
class Human{
    public void whoAmI(){
        System.out.println("I am human");
    }
}
class Student extends Human{
    public void whoAmI(){
        System.out.println("I am student");
    }
}
```

```
class GradStudent extends Student{
  public void whoAmI(){
    System.out.println("I am graduate student");
  }
}
```

```
public class Test {
    public static void testFnc(Human h){
        h.whoAmI();
    }

    public static void main(String[] args) {
        testFnc(new Human());
        testFnc(new Student());
        testFnc(new GradStudent());

        Human h  = new GradStudent();;
        h.whoAmI();

        Student s  = (Student)h;
        s.whoAmI();

        GradStudent gs = (GradStudent)s;
        gs.whoAmI();
    }
}
```

b) Write a program that will go through the items of a one dimensional array and check if a particular item is in that array using **enhanced for loop**. You can use a variable to store the item that you want to find. Take the following values as the initial values of the array: {1, 3, 2, 4, 1, 10, 19, 20, 5}. Use **main** method and appropriate class. **[2]**

Sample value to find	Sample output
1	Found
12	Not found