Lab Assignment 09

Inspiring Excellence

Course Code:	**CSE111**
Course Title:	**Programming Language II**
Topic:	**Inheritance & Method Overriding + Review**
Number of Tasks:	**11**

Task 1

Given the following classes, write the code for the **BBAStudent** class so that the following output is printed when we run the TestStudent class.

Driver Code	Output
public class Student{ private String name = "Just a Student"; private String department = "nothing"; public void updateDepartment(String dpt){ this.department = dpt; } public void updateName(String name){ this.name = name; } public void details(){ System.out.println("Name : " + name + " Department: " + department); } } **//Tester Class** public class TestStudent{ public static void main(String [] args){ BBAStudent b1 = new BBAStudent(); BBAStudent b2 = new BBAStudent("Humty Dumty"); BBAStudent b3 = new BBAStudent("Little Bo Peep"); b1.details(); System.out.println("1---------------"); b2.details(); System.out.println("2---------------"); b3.details(); } }	Name : Default Department: BBA 1--------------- Name : Humty Dumty Department: BBA 2--------------- Name : Little Bo Peep Department: BBA

Task 2

Given the following classes, write the code for the **Vehicle2010** class to print the following output when we run the Vehicle2010User class.

Driver Code	Output
public class Vehicle{ public int x; public int y; public void moveUp(){ y = y+1; } public void moveDown(){ y = y-1; } public void moveLeft(){ x = x-1; } public void moveRight(){ x = x+1; } public void position(){ System.out.println("("+ x + ","+ y + ")"); } } **//Tester Class** public class Vehicle2010User{ public static void main(String[] args){ Vehicle2010 car1 = new Vehicle2010(); car1.position(); car1.moveLowerLeft(); car1.position();; Vehicle2010 car2 = new Vehicle2010(); car2.position(); car2.moveUpperRight(); car2.position(); car2.moveLowerRight(); car2.position(); } }	(0,0) (-1,-1) (0,0) (1,1) (2,0)

Task 3

Design the **CheckingAccount** class derived from the Account class with appropriate attributes and properties so that the driver code can generate the output given below.

Driver Code	Output
public class Account{ public double balance = 0.0; public Account(double balance){ this.balance = balance; } public double showBalance(){ return balance; } } **//Tester Class** public class TestAccount{ public static void main(String [] args){ System.out.println("Total Checking Accounts: "+CheckingAccount.count); CheckingAccount c1 = new CheckingAccount(); System.out.println("Account Balance: " + c1.showBalance()); CheckingAccount c2 = new CheckingAccount(100.0); System.out.println("Account Balance: " + c2.showBalance()); CheckingAccount c3 = new CheckingAccount(200.0); System.out.println("Account Balance: " + c3.showBalance()); System.out.println("Total Checking Accounts: "+CheckingAccount.count); } }	Total Checking Accounts: 0 Account Balance: 0.0 Account Balance: 100.0 Account Balance: 200.0 Total Checking Accounts: 3

Task 4

Design the **Dog** and **Cat** class derived from the Animal class with appropriate attributes and properties so that the driver code can generate the output given below.

Driver Code

```
public class Animal {
    public String name;
    public int age;
    public String color;

    public Animal(String name, int age, String color) {
        this.name = name;
        this.age = age;
        this.color = color;
    }
    public void makeSound() {
        System.out.println("Animal makes a sound");
    }
    public String info() {
      return "Name: "+name+"\nAge: "+age+"\nColor: "+color+"\n";
    }
}
//Tester Class
public class AnimalTester {
    public static void main(String[] args) {
      Dog dog = new Dog("Buddy", 5, "Brown", "Bulldog");
      Cat cat = new Cat("Kitty", 3, "White", "Persian");
      System.out.println("1.========");
      System.out.println(dog.info());
      System.out.println("2.========");
      System.out.println(cat.info());
      System.out.println("3.========");
      dog.makeSound();
      System.out.println("4.========");
      cat.makeSound();
    }
}
```

Output

```
1.========
Name: Buddy
Age: 5
Color: Brown
Breed: Bulldog
2.========
Name: Kitty
Age: 3
Color: White
Breed: Persian
3.========
Brown color Buddy is barking
4.========
White color Kitty is meowing
```

Task 5

Implement the design of the **Smartphone** class so that the following output is produced. For simplicity, assume that a smartphone can have a maximum of 10 features.

Driver Code	Output
public class SmartPhoneTester{ public static void main(String[] args) { Smartphone s1 = new Smartphone(); System.out.println("1=========="); s1.addFeature("Display", "6.1 inch"); System.out.println("2=========="); s1.updateName("Samsung Note 20"); s1.addFeature("Display", "6.1 inch"); s1.printDetail(); System.out.println("3=========="); Smartphone s2 = new Smartphone("Iphone 12 Pro"); s2.addFeature("Display", "6.2 inch"); s2.addFeature("Ram", "6 GB"); System.out.println("4=========="); s2.printDetail(); s2.addFeature("Display", "Amoled panel"); s2.addFeature("Ram", "DDR5"); System.out.println("5=========="); s2.printDetail(); System.out.println("6=========="); } }	1========== Feature can not be added without phone name 2========== Phone Name: Samsung Note 20 Display: 6.1 inch 3========== 4========== Phone Name: Iphone 12 Pro Display: 6.2 inch Ram: 6 GB 5========== Phone Name: Iphone 12 Pro Display: 6.2 inch, Amoled panel Ram: 6 GB, DDR5 6==========

Task 6

Implement the **Bus** class so that the following output is produced.

Driver Code	Output
public class BusTester{ public static void main(String args[]){ Bus b1 = new Bus(4, "Jatrabari"); System.out.println("1--------------"); Bus b2 = new Bus(10, "Gazipur"); System.out.println("2--------------"); b1.addPassenger("Fahim", "Mirpur"); System.out.println("3--------------"); b1.addPassenger("Anika", "Jatrabari"); System.out.println("4--------------"); b1.addPassenger("Ali"); System.out.println("5--------------"); b1.addPassenger("Zafar"); System.out.println("6--------------"); b1.addPassenger("Mim", "Badda"); b1.addPassenger("Nowrin"); System.out.println("7--------------"); b1.addPassenger("Walid", "Jatrabari"); } }	Capacity: 4 Destination: Jatrabari 1-------------- Capacity: 10 Destination: Gazipur 2-------------- Sorry Fahim! The bus won't stop at Mirpur Use another bus. 3-------------- Anika is added to the bus. 4-------------- Ali is added to the bus. Ali will get off at the last stop 5-------------- Zafar is added to the bus. Zafar will get off at the last stop 6-------------- Sorry Mim! The bus won't stop at Badda Use another bus. Nowrin is added to the bus. Nowrin will get off at the last stop 7-------------- Bus is full.

Task 7

Implement the design of the **Account** class so that the following output is produced:

Driver Code	Output
public class AccountTester{ public static void main(String[] args) { System.out.println("Total account holders: " + Account.count); System.out.println("1==============="); Account p1 = new Account("Abdul",45,"Service Holder",500000); p1.addMoney(300000); p1.printDetails(); System.out.println("2==============="); Account p2 = new Account("Rahim",55,"Businessman",700000); p2.withdrawMoney(700000); p2.printDetails(); System.out.println("3==============="); Account p3 = new Account("Ashraf",62,"Govt.Officer",200000); p3.withdrawMoney(250000); p3.printDetails(); System.out.println("4==============="); System.out.println("Total account holders: " + Account.count); } }	Total account holders: 0 1=============== Name: Abdul Age: 45 Occupation: Service Holder Total Amount: 800000 2=============== Name: Rahim Age: 55 Occupation: Businessman Total Amount: 0 3=============== Insufficient money for withdrawal! Name: Ashraf Age: 62 Occupation: Govt.Officer Total Amount: 200000 4=============== Total account holders: 3

Task 8

Implement the **Student** class so that the following output is produced.

Driver Code	Output
public class StudentTester2{ public static void main(String[] args) { Student s1 = new Student("Naruto", "CSE"); System.out.println("1----------"); s1.individualInfo(); System.out.println("##########"); Student.totalInfo(); System.out.println("=========="); Student s2 = new Student("Sakura", "BBA"); System.out.println("2----------"); s2.individualInfo(); System.out.println("##########"); Student.totalInfo(); System.out.println("=========="); Student s3 = new Student("Shikamaru", "CSE"); System.out.println("3----------"); s3.individualInfo(); System.out.println("##########"); Student.totalInfo(); System.out.println("=========="); Student s4 = new Student("Deidara", "BBA"); System.out.println("4----------"); s4.individualInfo(); System.out.println("##########"); Student.totalInfo(); } }	Creating Student Number: 1 1---------- Naruto is from CSE department. Serial of Naruto among all students' is: 1 Serial of Naruto in CSE department is: 1 ########## Total Students: 1 Total CSE Students: 1 Total BBA Students: 0 ========== Creating Student Number: 2 2---------- Sakura is from BBA department. Serial of Sakura among all students' is: 2 Serial of Sakura in BBA department is: 1 ########## Total Students: 2 Total CSE Students: 1 Total BBA Students: 1 ========== Creating Student Number: 3 3---------- Shikamaru is from CSE department. Serial of Shikamaru among all students' is: 3 Serial of Shikamaru in CSE department is: 2 ########## Total Students: 3 Total CSE Students: 2 Total BBA Students: 1 ========== Creating Student Number: 4 4---------- Deidara is from BBA department. Serial of Deidara among all students' is: 4 Serial of Deidara in BBA department is: 2 ########## Total Students: 4 Total CSE Students: 2 Total BBA Students: 2

Task 9

```
1  public class Test1 {
2      int x = 2, y = 4, sum = 3;
3      int arr[] = {x, y, sum};
4      public void methodA(int x) {
5          arr[0] += methodB(y, this.x) + methodC(x);
6          System.out.println(x + " " + this.x + " " + sum);
7          arr[1] += this.x * (++y) / (sum % x);
8          System.out.println(y + " " + sum + " " + this.x);
9          arr[2] += methodC(x) + methodB(this.x, sum);
10         System.out.println(arr[0] + " " + arr[1] + " " + arr[2]);
11     }
12     public int methodB(int q, int n) {
13         int arr2[] = {7, 8};
14         int a = (arr2[0]++) - q;
15         int b = (++arr2[1]) - n;
16         return a + b;
17     }
18     public int methodC(int z) {
19         z = sum + methodB(x, sum) - z;
20         return z/2;
21     }
22 }
```

```
public class Tester1{
    public static void main(String [] args){
        Test1 t1 = new Test1();
        t1.methodA(7);
    }
}
```

Outputs		

Task 10

```
1  public class Test3 {
2      int x = 2, y = 4, z = 5;
3      double p = 0.0;
4      public void methodA(int x, int m) {
5          this.x = methodC(this.x);
6          p = x + this.x % m * 3.0;
7          y = y + methodB(x, this.x);
8          System.out.println(this.x +" " + x + y + " " + p) ;
9      }
10     public int methodB(int q, int n) {
11         int arr[] = {3,4,5};
12         arr[0] = arr[0] + this.x + n;
13         arr[1] = q + arr[1];
14         System.out.println(arr[0] +" " + arr[1] + " " + arr[2]) ;
15         return arr[1] + arr[2];
16     }
17     public int methodC(int y) {
18         if(y % 2 == 0) {
19             int temp = methodB(2, y);
20             return temp;
```

```
21          }
22          else{
23              return 4;
24          }
25      }
26  }
```

Driver Code	Output		
Test3 t3 = new Test3(); t3.methodA(2,3); t3.methodB(5,4);			

Task 11

```
1   public class Quiz3A{
2     public static int temp = 4;
3     public static int y;
4     public int sum;
5     public Quiz3A(){
6       int y = 7;
7       y = temp - 1;
8       sum = Quiz3A.temp + 1 + y;
9       temp+=2;
10    }
```

```
11    public Quiz3A(int k){
12      temp = temp++;
13      sum = ++temp + k;
14      Quiz3A.y = (sum++) - 1;
15      System.out.println(Quiz3A.y+" "+temp+" "+y);
16    }
17    public int methodB(int m, int n){
18      int x = 0;
19      y = this.y + m + (++temp);
20      x = x + 2 +  n;
21      sum = sum + x + y;
22      System.out.println(x + " " + this.y+ " " + sum);
23      return sum;
24    }
25  }
```

Driver Code	Output		
public class Tester2{ public static void main(String args[]){ Quiz3A a1 = new Quiz3A(); a1.methodB(1,2); Quiz3A a2 = new Quiz3A(3); a2.methodB(2,4); a1.methodB(2,1); } }			