Lab Assignment 07

BRAC UNIVERSITY

Inspiring Excellence

Course Code:	**CSE111**
Course Title:	**Programming Language II**
Topic:	**Inheritance**
Number of Tasks:	**12 (Classwork: 06, Homework: 06)**

[Submit all the Coding Tasks (Homework: Task 1 to 5) in the Google Form shared on buX before the next lab.]

[You are not allowed to change the driver codes of any of the tasks]

CLASSWORK

Task 1

Complete the class **Circle** so that the desired outputs are generated properly.

Given Code	Expected Output

public class shapeTester {
  public static void main(String[] args) {
    Shape s = new Shape();
    s.name = "Mobius Strip";
    s.color = "Blue";
    s.displayInfo();
    System.out.println("1=======================");
    Circle c = new Circle();
    System.out.println("2=======================");
    c.name = "Circle";
    c.color = "Red";
    c.radius = 5;
    c.displayInfo();
    System.out.println("3=======================");
    c.area();
  }
}
 	
public class Shape {
  public String name;
  public String color;
 	
  public void displayInfo() {
    System.out.printf("Name: %s\nColor: %s\n", name, color);
  }
}
 	
public class Circle extends Shape {
    //Your Code Here
}
	Name: Mobius Strip
	Color: Blue
	1=======================
	2=======================
	Name: Circle
	Color: Red
	3=======================
	Area of Red Circle: 78.54

Task 2

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 and Parent 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();
  }
}

public class Student{
  private String name = "Just a Student";
  private String department = "nothing";

  public void setDepartment(String dpt){
    this.department = dpt;
  }
  public void setName(String name){
    this.name = name;
  }
  public void details(){
    System.out.println("Name : " + name + " Department: "
+ department);
  }
}
```

Output

```
Name: Default Department: BBA
1---------------
Name: Humty Dumty Department: BBA
2---------------
Name: Little Bo Peep Department: BBA
```

Task 3

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

Driver Code and Parent Class

```
public class Vehicle2010User{
  public static void main(String[] args){
    Vehicle2010 car1 = new Vehicle2010();
    System.out.println(car1.getPosition());
    car1.moveLowerLeft();
    System.out.println(car1.getPosition());
    Vehicle2010 car2 = new Vehicle2010();
    System.out.println(car2.getPosition());
    car2.moveUpperRight();
    System.out.println(car2.getPosition());
    car2.moveLowerRight();
    System.out.println(car2.getPosition());
  }
}

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 String getPosition(){
    return "("+ x + ","+ y + ")";
  }
}
```

Output

```
(0,0)
(-1,-1)
(0,0)
(1,1)
(2,0)
```

Task 4

Given the following classes, write the code for the **Cricket_Tournament** and the **Tennis_Tournment** classes derived from **Tournament** so that the following output is generated.

<table>
<tr><th>Given Code</th><th>Expected Output</th></tr>
<tr><td>

```
public class TournamentTester {
  public static void main(String[] args) {
    Cricket_Tournament ct1 = new Cricket_Tournament();
    System.out.println(ct1.info());
    System.out.println("-----------------------");

    Cricket_Tournament ct2 = new Cricket_Tournament("IPL", 10, "t20");
    System.out.println(ct2.info());
    System.out.println("-----------------------");

    Tennis_Tournament tt = new Tennis_Tournament("Roland Garros", 128);
    System.out.println(tt.info());
  }
}

public class Tournament {
  private String name;

  public Tournament() {
    this.name = "Default";
  }

  public Tournament(String name) {
    this.name = name;

  }

  public String getName(){
    return "Tournament Name: "+name;
  }
}
```

</td><td>

```
Cricket Tournament Name: Default
Number of Teams: 0
Type: No type
-----------------------
Cricket Tournament Name: IPL
Number of Teams: 10
Type: t20
-----------------------
Tennis Tournament Name: Roland Garros
Number of Players: 128
```

</td></tr>
</table>

Task 5

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 and Parent Class

```
public class AnimalTester1 {
  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.dogInfo());
    System.out.println("2.========");
    System.out.println(cat.catInfo());
    System.out.println("3.========");
    dog.makeSound();
    System.out.println("4.========");
    cat.makeSound();
  }
}

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 String info() {
    return "Name: "+name+"\nAge: "+age+"\nColor:
"+color+"\n";
  }
}
```

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 6

```
public class A{
  public int temp = 4;
  public int sum = 1;
  public int y = 2;
  public void methodA(int m, int n){
    int x = 0;
    y = y + m + (temp++);
    x = x + 2 +  n;
    sum = sum + x + y;
    System.out.println(x + " " + y+ " " + sum);
  }
}
public class B extends A {
  public int x = 6;
  public void methodB(int m, int n){
   int  y = 0;
   y = ++y + this.y;
   x = this.y + 2 + temp;
   methodA(x, y);
   sum = x + y + super.sum;
   System.out.println(x + " " + y+ " " + sum);
  }
}
```

```
A a1 = new A();
a1.methodA(1, 1);
B b1 = new B();
b1.methodB(1, 2);
```

x	y	sum

HOMEWORK

Task 1

Complete the class **Cow** so that the desired outputs are generated properly.

Given Code

```
public class AnimalTester2 {
  public static void main(String args[]){
    Animal a1 = new Animal();
    System.out.println("1-------------");
    a1.details();
    System.out.println("2-------------");
    Cow c1  = new Cow();
    c1.name = "Pammy";
    System.out.println("3-------------");
    System.out.println("Name: " + c1.getName());
    c1.details();
    System.out.println("4-------------");
    c1.updateSound("Moo");
    System.out.println("5-------------");
    c1.details();
  }
}

public class Animal{
  public int legs = 4;
  public String sound = "Not defined";

  public void details(){
    System.out.println("Legs: "+legs);
    System.out.println("Sound: "+sound);
  }
}

public class Cow extends Animal{
    //Your Code Here
}
```

Expected Output

```
1-------------
Legs: 4
Sound: Not defined
2-------------
The cow says hello!
3-------------
Name: Pammy
Legs: 4
Sound: Not defined
4-------------
5-------------
Legs: 4
Sound: Moo
```

Task 2

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 and Parent Class

```
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);
  }
}
```

Output

```
Total Checking Accounts: 0
Account Balance: 0.0
Account Balance: 100.0
Account Balance: 200.0
Total Checking Accounts: 3
```

Task 3

Given the following classes, write the code for the **Book** and the **CD** class so that the following output is printed.

Driver Code and Parent Class

```java
public class Tester6 {
  public static void main(String[] args) {
    Book book = new Book(1, "The Alchemist", 500, "97806",
"HarperCollins");
    System.out.println(book.printDetail());
    System.out.println("-----------------------");

    CD cd = new CD(2, "Shotto", 300, "Warfaze", 50, "Hard Rock");
    System.out.println(cd.printDetail());
  }
}

class Product {
  private int id;
  private String title;
  private int price;

  public Product(int id, String title, int price) {
    this.id = id;
    this.title = title;
    this.price = price;
  }

  public String getIdTitlePrice() {
    return "ID: " + id + " Title: " + title + " Price: " + price;
  }
}
```

Expected Output

```
ID: 1 Title: The Alchemist Price: 500
ISBN: 97806 Publisher: HarperCollins
-----------------------
ID: 2 Title: Shotto Price: 300
Band: Warfaze Duration: 50 minutes
Genre: Hard Rock
```

Task 4

Design the **ComplexNumber** class with the necessary property to produce the output from the given driver code.

Driver Code and Parent Class

```
public class ComplexNumberTester {
    public static void main(String[] args) {
        ComplexNumber cn1 = new ComplexNumber();
        System.out.println(cn1.details());
        System.out.println("----------------");
        ComplexNumber cn2 = new ComplexNumber(5.0, 7.0);
        System.out.println(cn2.details());
    }
}

public class RealNumber {
    public double realValue;
    public RealNumber() {
        this(0.0);
    }
    public RealNumber(double realValue) {
        this.realValue = realValue;
    }
    public String getReal(){
        return "RealPart: " + realValue;
    }
}
```

Output

```
RealPart: 1.0
ImaginaryPart: 1.0
----------------
RealPart: 5.0
ImaginaryPart: 7.0
```

Task 5

Given the following classes, write the code for the **CSEStudent** class derived from **Student** so that the following output is generated.

Given Code

```
public class StudentTester{
  public static void main (String args[]){
    CSEStudent.details();
    System.out.println("1--------------");
    CSEStudent s1 =  new CSEStudent("Bob", 23);
    s1.info();
    System.out.println("2--------------");
    CSEStudent s2 =  new CSEStudent("Don", 33);
    s2.info();
    System.out.println("3--------------");
    s1.addLabBasedCourse("CSE220");
    s1.addLabBasedCourse("CSE221");
    System.out.println("4--------------");
    s1.info();
    System.out.println("5--------------");
    CSEStudent.details();
    System.out.println("6--------------");
    s1.addLabBasedCourse("CSE230");
    System.out.println("7--------------");
    s1.info();
    System.out.println("8--------------");
    s2.addLabBasedCourse("CSE110");
    s2.info();
  }
}

class Student{
  public String name;
  public int id;
  public String courses = "";

  public Student(String n, int i){
    name = n;
    id = i;
  }

  public void info(){
    System.out.println("Name: "+name);
    System.out.println("ID: "+id);
    System.out.println("Courses: "+courses);
  }
}
```

Expected Output

```
Total CSE Students: 0
Available Lab Based Courses:
CSE110 CSE111 CSE220 CSE221
1--------------
Name: Bob
ID: 23
Courses:
2--------------
Name: Don
ID: 33
Courses:
3--------------
4--------------
Name: Bob
ID: 23
Courses: CSE220 CSE221
5--------------
Total CSE Students: 2
Available Lab Based Courses:
CSE110 CSE111 CSE220 CSE221
6--------------
It is not a lab based course!
7--------------
Name: Bob
ID: 23
Courses: CSE220 CSE221
8--------------
Name: Don
ID: 33
Courses: CSE110
```

Task 6

```
public class A{
 public int m = 14, n = -1, p = 3;
 public static int temp = 11;
 public void methodA(int m, int n){
   int x = 0;
   p = this.p + m + (this.temp++);
   x = x + 2 +  n;
   this.n = n + B.x + p;
   System.out.println(this.m + " " + p+ " " + this.n);
 }
}
public class B extends A {
 public static int x = 6;
 public void methodB(int x, int y){
  y = ++y + this.p + n;
  super.p = x + 2 + temp;
  methodA(temp, x);
  B.x = this.x + x + super.temp;
  System.out.println(B.x + " " + y+ " " + A.temp);
 }
}
```

B b1 = new B();
b1.methodB(3, 5);
A a1 = new A();
a1.methodA(12, -9);
	x	y	sum

Ungraded Tasks (Optional)

(You don’t have to submit the ungraded tasks)

Task 1

Given the following classes, write the code for the **Player** and the **Manager** classes derived from SportsPerson class so that the following output is printed. To calculate the match earnings use the following formula:

1. Player: (total_goal * 1000) + (total_match * 10)
2. Manager: match_win * 1000

Given Code

```
public class PlayerTester {
  public static void main(String[] args) {
    Player playerOne = new Player("Al-Nassr", "Ronaldo",
"Striker", 25, 32);
    playerOne.calculateRatio();
    playerOne.printDetails();
    System.out.println("----------------");
    Manager managerOne = new Manager("Real Madrid",
"Zidane", "Manager", 25);
    managerOne.printDetails();
  }
}

class SportsPerson {
  private String team;
  private String name;
  public String role;
  public double earningPerMatch;

  public SportsPerson(String teamName, String name, String
role){
    this.team = teamName;
    this.name = name;
    this.role = role;
    this.earningPerMatch = 0;
  }

  public String getNameTeam() {
    return "Name: " + name + ", Team Name: " + team;
  }
}
```

Expected Output

```
Name: Ronaldo, Team Name: Al-Nassr
Team Role: Striker
Total Goal: 25, Total Played: 32
Goal Ratio: 0.78
Match Earning: 25320K
----------------
Name: Zidane, Team Name: Real
Madrid
Team Role: Manager
Total Win: 25
Match Earning: 25000K
```