

Lab Assignment 08



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
<pre>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 }</pre>	<pre>Name: Mobius Strip Color: Blue 1===== 2===== Name: Circle Color: Red 3===== Area of Red Circle: 78.54</pre>

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 and Parent Class	Output
<pre>public class Vehicle2010User{ public static void main(String[] args){ Vehicle2010 car1 = new Vehicle2010(); System.out.println(car1); car1.moveLowerLeft(); System.out.println(car1); Vehicle2010 car2 = new Vehicle2010(); System.out.println(car2); car2.moveUpperRight(); System.out.println(car2); car2.moveLowerRight(); System.out.println(car2); } } 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 toString(){ return "("+ x + ","+ y + ")"; } }</pre>	<pre>(0,0) (-1,-1) (0,0) (1,1) (2,0)</pre>

Task 3

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

Driver Code and Parent Class	Output
<pre>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); } } //Tester Class public class TestStudent{ public static void main(String [] args){ BBASStudent b1 = new BBASStudent(); BBASStudent b2 = new BBASStudent("Humty Dumty"); BBASStudent b3 = new BBASStudent("Little Bo Peep"); b1.details(); System.out.println("1-----"); b2.details(); System.out.println("2-----"); b3.details(); } }</pre>	<pre>Name: Default Department: BBA 1----- Name: Humty Dumty Department: BBA 2----- Name: Little Bo Peep Department: BBA</pre>

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 and Parent Class	Output
<pre>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"; } } public class AnimalTester2 { 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(); } }</pre>	<pre>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</pre>

Task 5

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

Given Code	Expected Output
<pre>public class TournamentTester { public static void main(String[] args) { Cricket_Tournament ct1 = new Cricket_Tournament(); System.out.println(ct1); System.out.println("-----"); Cricket_Tournament ct2 = new Cricket_Tournament("IPL", 10, "t20"); System.out.println(ct2); System.out.println("-----"); Tennis_Tournament tt = new Tennis_Tournament("Roland Garros", 128); System.out.println(tt); } } public class Tournament { private String name; public Tournament() { this.name = "Default"; } public Tournament(String name) { this.name = name; } public String getName(){ return this.name; } @Override public String toString(){ return "Tournament Name: "+name; } }</pre>	<pre>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</pre>

Task 6

1	public class A{
2	public int temp = 4;
3	public int sum = 1;
4	public int y = 2;
5	public A(){
6	y = temp - 2;
7	sum = temp + 3;
8	temp-=2;
9	}
10	public void methodA(int m, int n){
11	int x = 0;
12	y = y + m + (temp++);
13	x = x + 2 + n;
14	sum = sum + x + y;
15	System.out.println(x + " " + y+ " " + sum);
16	}
17	}
18	public class B extends A {
19	public int x;
20	public B(){
21	y = temp + 3 ;
22	sum = 3 + temp + 2;
23	temp-=1;
24	}
25	public B(B b){
26	sum = b.sum;
27	x = b.x;
28	}
29	public void methodB(int m, int n){
30	int y = 0;
31	y = y + this.y;
32	x = this.y + 2 + temp;
33	methodA(x, y);
34	sum = x + y + super.sum;
35	System.out.println(x + " " + y+ " " + sum);
36	}
37	}

A a1 = new A();	x	y	sum
B b1 = new B();			
B b2 = new B(b1);			
a1.methodA(1, 1);			
b1.methodA(1, 2);			
b2.methodB(3, 2);			

HOMEWORK

Task 1

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

Given Code	Expected Output
<pre>public class AnimalTester{ public static void main(String args[]){ Animal a1 = new Animal(); System.out.println("1-----"); a1.details(); System.out.println("2-----"); Dog d1 = new Dog(); d1.name = "Pammy"; System.out.println("3-----"); System.out.println("Name: " + d1.getName()); d1.details(); System.out.println("4-----"); d1.updateSound("Bark"); System.out.println("5-----"); d1.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 Dog extends Animal{ //Your Code Here }</pre>	<pre>1----- Legs: 4 Sound: Not defined 2----- The dog says hello! 3----- Name: Pammy Legs: 4 Sound: Not defined 4----- 5----- Legs: 4 Sound: Bark</pre>

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	Output
<pre>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); } }</pre>	<pre>Total Checking Accounts: 0 Account Balance: 0.0 Account Balance: 100.0 Account Balance: 200.0 Total Checking Accounts: 3</pre>

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	Expected Output
<pre>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; } }</pre>	<pre>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</pre>

Task 4

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

Given Code	Expected Output
<pre>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); } }</pre>	<pre>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</pre>

Task 5

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

Driver Code and Parent Class	Output
<pre>public class RealNumber { public double realValue; public RealNumber() { this(0.0); } public RealNumber(double realValue) { this.realValue = realValue; } public String toString(){ return "RealPart: " + realValue; } } public class ComplexNumberTester { public static void main(String[] args) { ComplexNumber cn1 = new ComplexNumber(); System.out.println(cn1); System.out.println("-----"); ComplexNumber cn2 = new ComplexNumber(5.0, 7.0); System.out.println(cn2); } }</pre>	<pre>RealPart: 1.0 ImaginaryPart: 1.0 ----- RealPart: 5.0 ImaginaryPart: 7.0</pre>

Task 6

1	public class A{
2	public static int temp = 3;
3	public int sum;
4	public int y;
5	public A(){
6	y = temp - 1;
7	sum = temp + 2;
8	temp-=2;
9	}
10	public void methodA(int m, int [] n){
11	int x = 0;
12	y = y + m + (temp++);
13	x = x + 2 + (++n[0]);
14	sum = sum + x + y;
15	n[0] = sum + 2;
16	System.out.println(x + " " + y+ " " + sum);
17	}
18	}
19	class B extends A {
20	public static int x = 1;
21	public B(){
22	y = temp + 1 ;
23	x = 3 + temp + x;
24	temp-=2;
25	}
26	public B(B b){
27	sum = b.sum + super.sum;
28	x = b.x + x;
29	}
30	public void methodB(int m, int n){
31	int [] y = {0};
32	super.y = y[0] + this.y + m;
33	x = super.y + 2 + temp - n;
34	methodA(x, y);
35	sum = x + y[0] + super.sum;
36	System.out.println(x + " " + y[0]+ " " + sum);
37	}
38	}

<pre>int x[] = {23}; A a1 = new A(); B b1 = new B(); B b2 = new B(b1); a1.methodA(1, x); b2.methodB(3, 2); a1.methodA(1, x);</pre>	Output		

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: $(\text{total_goal} * 1000) + (\text{total_match} * 10)$
2. Manager: $\text{match_win} * 1000$

Given Code	Expected Output
<pre>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; } }</pre>	<pre>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</pre>

Task2

1	public class A {
2	public static int temp = 4;
3	public static int x = -10;
4	public int sum = 0;
5	public int y = 0;
7	public A() {
8	y = temp - 2;
9	sum = temp + 1;
10	temp -= 2;
11	}
13	public void methodA(int m, int n) {
14	int x = 0;
15	y = y + m + (temp++);
16	x = x + 1 + n;
17	sum = sum + x + y;
18	System.out.println(x + " " + y + " " + sum);
19	}
20	}
22	public class B extends A {
23	public static int x = 0;
24	public int sum = -6;
25	public B() {
26	sum = 0;
27	y = temp + 3;
28	super.sum = 3 + temp + 2;
29	temp -= 2;
30	}
31	public B(B b) {
33	sum = b.sum + super.sum;
34	x = b.x;
35	b.methodB(2, 3);

