Lab Assignment 07

Inspiring Excellence

Course Code:	**CSE111**
Course Title:	**Programming Language II**
Topic:	**Constructor, Constructor Overloading, Method Overloading**
Number of Tasks:	**11**

[Submit all the Coding Tasks in the Google Form shared on buX before the next lab. Submit the Tracing Tasks handwritten to your Lab Instructors at the beginning of the lab]

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

Task 1

Design the **Student** class in such a way that it produces the following output.

Driver Code	Expected Output

`public class StudentTester{`
`  public static void main(String[] args){`
`  Student s1 = new Student("Harry", "CSE");`
`  System.out.println(s1.name);`
`  s1.updateName("Harry Potter");`
`  System.out.println(s1.accessName());`
`  System.out.println(s1.prog);`
`  s1.updateProgram("CS");`
`  String prog = s1.accessProgram();`
`  System.out.println(prog);`
`  }`
`}`
	`Harry`
	`Harry Potter`
	`CSE`
	`CS`

Task 2

Design the **Toy** class in such a way that it produces the following output

Driver Code	Expected Output

`public class ToyTester{`
`  public static void main(String[] args){`
`  Toy t1 = new Toy("Car", 230);`
`  System.out.println("1================");`
`  t1.updatePrice(250);`
`  System.out.println("2================");`
`  System.out.println(t1.name);`
`  t1.showPrice();`
`  System.out.println("3================");`
`  Toy t2 = new Toy("Robot", 450);`
`  System.out.println("4================");`
`  t2.updateName("Autobot");`
`  System.out.println("5================");`
`  System.out.println(t2.name);`
`  t2.showPrice();`
`  }`
`}`
	`A new toy has been made!`
	`1================`
	`2================`
	`Car`
	`price: 250 Taka`
	`3================`
	`A new toy has been made!`
	`4================`
	`Changing old name: Robot`
	`new name: Autobot`
	`5================`
	`Autobot`
	`price: 450 Taka`

Task 3

Design the **Parcel** class in such a way that it produces the following output.

NOTE: For the method ***calcFee()***, if the delivery location is ***Dhanmondi***, then the location charge will be 50 taka or else it'll be free. Also, while calculating total fee, if the product weight is 0 the total_fee would also be 0.

Formula: fee = (weight * 20) + *location_charge (if any)*

Driver Code	Expected Output

`public class ParcelDriver {`
`  public static void main(String[] args){`
`  Parcel p1 = new Parcel();`
`  p1.printDetails();`
`  p1.name = "Spongebob";`
`  p1.printDetails();`
`  System.out.println("1**********");`
`  Parcel p2 = new Parcel("Bob the Builder");`
`  p2.weight = 15;`
`  p2.calcFee("Gulshan");`
`  p2.printDetails();`
`  System.out.println("2**********");`
`  p2.addWeight(25);`
`  p2.calcFee("Banani");`
`  p2.printDetails();`
`  System.out.println("3**********");`
`  Parcel p3 = new Parcel("Dora the Explorer", 10);`
`  p3.addWeight(15);`
`  p3.calcFee("Dhanmondi");`
`  p3.printDetails();`
`  }`
`}`
	`Set name first`
	`Name: Spongebob`
	`Total Weight: 0`
	`Total Fee: 0.0`
	`1**********`
	`Name: Bob the Builder`
	`Total Weight: 15`
	`Total Fee: 300.0`
	`2**********`
	`Updated Weight: 40`
	`Name: Bob the Builder`
	`Total Weight: 40`
	`Total Fee: 800.0`
	`3**********`
	`Updated Weight: 25`
	`Name: Dora the Explorer`
	`Total Weight: 25`
	`Total Fee: 550.0`

Task 4

Design the **Shape2D** class in such a way that it produces the following output.

Driver Code	Expected Output

`public class Shape2DTester {`
`  public static void main(String[] args) {`
`  Shape2D sq = new Shape2D(5);`
`  System.out.println("---------1----------");`
`  sq.area();`
`  System.out.println("---------2----------");`
`  shape2D rectangle = new shape2D(5,6);`
`  System.out.println("---------3----------");`
`  rectangle.area();`
`  System.out.println("---------4----------");`
`  shape2D tri1 = new shape2D(5,6,"Triangle");`
`  System.out.println("---------5----------");`
`  tri1.area();`
`  System.out.println("---------6----------");`
`  shape2D tri2 = new shape2D(5,6,7);`
`  System.out.println("---------7----------");`
`  tri2.area();`
`  System.out.println("---------8----------");`
`  }`
`}`
	`A Square has been created with length: 5`
	`---------1----------`
	`The area of the Square is: 25.0`
	`---------2----------`
	`A Rectangle has been created with length: 5 and breadth:  6`
	`---------3----------`
	`The area of the Rectangle is: 30.0`
	`---------4----------`
	`A Triangle has been created with height: 5 and base:  6`
	`---------5----------`
	`The area of the Triangle is: 15.0`
	`---------6----------`
	`A Triangle has been created with the following sides: 5, 6, 7`
	`---------5----------`
	`The area of the Triangle is: 14.69`

Task 5

Write "**Book**" class to show the following output :

Driver Code	Output

`public class BookTester {`
`  public static void main(String[] args) {`
`  System.out.println("< —--—----1—--------->");`
`  Book b1 = new Book("The Alchemist");`
`  b1.displayDetails();`
`  System.out.println("< —--—----2—--------->");`
`  Book b2 = new Book("1984", "George Orwell");`
`  b2.displayDetails();`
`  System.out.println("< —--—----3—--------->");`
`  Book b3 = new Book("To Kill a Mockingbird", "Harper Lee", 300);`
`  b3.displayDetails();`
	`< —--—----1—--------->`
	`Title: The Alchemist`
	`< —--—----2—--------->`
	`Title: 1984, Author: George Orwell`
	`< —--—----3—--------->`
	`Title: To Kill a Mockingbird, Author: Harper Lee, Price: 300`
	`< —--—----4—--------->`
	`Title: The Alchemist, Price: 250`
	`< —--—----5—--------->`
	`Title: 1984, Author: Orwell, Price: 350`

```
    System.out.println("< —--—----4—--------->");
    b1.setDetails(250);
    b1.displayDetails();
    System.out.println("< —--—----5—--------->");
    b2.setDetails("Orwell", 350);
    b2.displayDetails();
  }
}
```

Task 6

Write "**Product**" class to show the following output :
Note: Make sure to use proper ***Encapsulation concepts*** **for the setter & getter methods**. All the attributes should have **Private** access.

Driver Code	Output

`public class ProductTester{`
`  public static void main(String[] args) {`
`    System.out.println("< —--—----1—--------->");`
`    Product product1 = new Product();`
`    product1.displayInfo();`
`    System.out.println();`
`    System.out.println("< —--—----2—--------->");`
`    Product product2 = new Product("Laptop", 1200.00);`
`    product2.setQuantity(10);`
`    product2.displayInfo(true);`
`    System.out.println("< —--—----3—--------->");`
`    System.out.println("Retrieved Price: $" +`
`product2.getPrice());`
`    System.out.println("Retrieved Quantity: " +`
`product2.getQuantity());`
`      }`
`}`
	`< —--—----1—--------->`
	`Product Name: Unknown`
	`Price: $0.0`
	`< —--—----2—--------->`
	`Product Name: Laptop`
	`Price: $1200.0`
	`Quantity: 10`
	`< —--—----3—--------->`
	`Retrieved Price: $1200.0`
	`Retrieved Quantity: 10`

Task 7

Write "**Student**" class to show the following expected outputs

Note:

- Make sure to use proper ***Encapsulation concepts*** **for the setter methods**. All the attributes should have **Private** access.
- A student can't take any course until the CGPA is set.
- A student cannot take more than 4 courses.
- A student with CGPA below 3 cannot take more than 3 courses.

Driver Code	Expected Output

`public class StudentDriver {`
`  public static void main(String[] args){`
`  System.out.println("-----------------");`
`  Student student1 = new Student(12345678);`
`  student1.addCourse("CSE110");`
`  student1.setCG(2.5);`
`  student1.addCourse("CSE110");`
`  student1.addCourse("ENG101");`
`  student1.showAdvisee();`
`  System.out.println("-----------------");`
`  student1.rmAllCourse();`
`  student1.showAdvisee();`
`  System.out.println("-----------------");`
`  student1.setID(54652365);`
`  String[] courses = {"SOC101","CSE111","ENG102"};`
`  student1.addCourse(courses);`
`  student1.showAdvisee();`
`  System.out.println("-----------------");`
`  student1.addCourse("CSE230");`
`  student1.showAdvisee();`
`  System.out.println("-----------------");`
`  Student student2 = new Student(975738383,3.7);`
`  String[] courses2 =`
`{"CSE220","PHY112","MAT120","BUS101","CHN101"};`
`  student2.addCourse(courses2);`
`  student2.showAdvisee();`
`  }`
`}`
	`-----------------`
	`Failed to add CSE110`
	`Set CG first`
	`Student ID: 12345678, CGPA: 2.5`
	`Added courses are:`
	`CSE110 ENG101`
	`-----------------`
	`Student ID: 12345678, CGPA: 2.5`
	`No courses added.`
	`-----------------`
	`Student ID: 54652365, CGPA: 2.5`
	`Added courses are:`
	`SOC101 CSE111 ENG102`
	`-----------------`
	`Failed to add CSE230`
	`CG is low. Can't add more than 3`
	`courses.`
	`Student ID: 54652365, CGPA: 2.5`
	`Added courses are:`
	`SOC101 CSE111 ENG102`
	`-----------------`
	`Failed to add CHN101`
	`Maximum 4 courses allowed.`
	`Student ID: 975738383, CGPA: 3.7`
	`Added courses are:`
	`CSE220 PHY112 MAT120 BUS101`

Task 8

Design "**ABCServer**" class to show the following output :

Driver Class	Output

public class ABCServerTester{
 public static void main (String args []){
  ABCServer server1 = new ABCServer();
  server1.details();
  System.out.println("---------------");
  ABCServer server2 = new ABCServer("Heroes Reborn",6);
  server2.details();
  System.out.println("---------------");
  server2.addMembers("Edward");
  server2.addMembers("William");
  System.out.println("----------------");
  server2.details();
  System.out.println("----------------");
  server2.addMembers("John", "Hero's Mentor");
  server2.addMembers("Albert", "Thunderstrike");
  server2.addMembers("Max", "Radiant Avenger");
  System.out.println("----------------");
  server2.details();
  System.out.println("----------------");
  server2.addMembers("Daniel");
  server2.addMembers("Donal", "Valor Knight");
  System.out.println("----------------");
  server2.details();
 }
}
	Server Name: Default
	Member Capacity: 10
	Total Members: 0
	Members:
	--------------- 	
	Server Name: Heroes Reborn
	Member Capacity: 6
	Total Members: 0
	Members:
	--------------- 	
	Rising Hero is added.
	Rising Hero is added.
	---------------- 	
	Server Name: Heroes Reborn
	Member Capacity: 6
	Total Members: 2
	Members:
	Name:Role --> Edward:Rising Hero
	Name:Role --> William:Rising Hero
	---------------- 	
	Hero's Mentor is added.
	Thunderstrike is added.
	Radiant Avenger is added.
	---------------- 	
	Server Name: Heroes Reborn
	Member Capacity: 6
	Total Members: 5
	Members:
	Name:Role --> Edward:Rising Hero
	Name:Role --> William:Rising Hero
	Name:Role --> John:Hero's Mentor
	Name:Role --> Albert:Thunderstrike
	Name:Role --> Max:Radiant Avenger
	---------------- 	
	Rising Hero is added.
	Sorry, maximum capacity exceed!
	---------------- 	
	Server Name: Heroes Reborn
	Member Capacity: 6
	Total Members: 6

	Members: Name:Role --> Edward:Rising Hero Name:Role --> William:Rising Hero Name:Role --> John:Hero's Mentor Name:Role --> Albert:Thunderstrike Name:Role --> Max:Radiant Avenger Name:Role --> Daniel:Rising Hero

Task 9

```
1  public class Trace1{
2       public int[] q;
3       public int x, y;
4       public Trace1(int[] p){
5            this.q = p;
6            System.out.println(q[1]+" "+q[2]+" "+q[3]);
7       }
8       public int m2(int a, int b){
9            x = b++;
10           y = ++a/x;
11           q[x] = b + q[x];
12           if(b%2==0){
13                q[b] = x;
14                System.out.println(q[a]+" "+y+" "+x);
15                this.m1(b,a);
16           }
17           else{
18                System.out.println(x+" "+y+" "+q[x]);
```

```
19              }
20              return x+y;
21          }
22          public void m1(int x, double y){
23              this.y = (int)y;
24              System.out.println(q[x]+" "+(++x)+" "+y );
25              m2(this.y,x-1);
26          }
27  }
```

DRIVER CODE	OUTPUTS		

public class Main {
  public static void main(String[] args){
  int[] arr = {5,3,9,2,1};
  Trace1 t1 = new Trace1(arr);
  int x = t1.m2(7,2);
  System.out.println(arr[0]+" "+x+" "+arr[4]);
  t1.m1(2,7);
  }
}			

Task 10

```
1   public class Maze{
```

```
2         public int x;
3         public void methodA(){
4              int m = 0, x = 9;
5             m = methodB(m-3)+x;
6             this.x = ++x;
7              System.out.println(this.x+" "+m);
8              methodB(x,m);
9             System.out.println(x+" "+(m+this.x));
10            methodB(m);
11        }
12        public int methodB(int y){
13             x=y*y;
14            System.out.println(x+" "+y);
15             return x-11;
16        }
17        public void methodB(int z, int x){
18             z=z-2;
19            x=this.x-2*x;
20            System.out.println(z+" "+this.x);
21        }
22    }
```

DRIVER CODE	OUTPUTS	
public class MazeTester{ public static void main(String args []){		

```
    Maze m1 = new Maze();
    m1.methodA();
  }
}
```


Task 11

```
1  public class Puzzle{
2       public int x,z;
3       public Puzzle(int x){
4            this.x = x;
5       }
6       public Puzzle(int x, int z){
7            this.x = x;
8            this.z = z;
9       }
10      public void methodA(){
11           z=x+methodB(x);
12           System.out.println(x+" "+z);
13           methodB(x,z);
14      }
15      public int methodB(int y){
16           x=y+x;
17           System.out.println(x+" "+y);
18           return x+3;
19      }
```

20	public void methodB(int z, int x){
21	z=this.z+1;
22	x=x+1;
23	System.out.println(z+" "+x);
24	}
25	}

DRIVER CODE	OUTPUTS	

public class PuzzleTester{
public static void main(String[]args){
Puzzle p = new Puzzle(5,8);
Puzzle p1 = new Puzzle(8);
p.methodA();
System.out.println(p.methodB(7)+" "+p1.methodB(7));
}
}		