Lab Assignment 05

Inspiring Excellence

Course Code:	**CSE111**
Course Title:	**Programming Language II**
Topic:	**Reference Passing & Multiclass Problem (Continued)**
Number of Tasks:	**9 (Classwork: 03, Homework: 06)**

[Submit all the Coding Tasks (Homework: Task 1 to 4) in the Google Form shared on buX before the next lab. Submit the Tracing Tasks (Homework: Task 5 to 6) 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]

CLASSWORK

Task 1

Carefully observe the following driver code and expected output. Write the java code for the **MFS** Class and **Card** Class both. Consider that there can be at most 1 card per MFS account.

Tester Class	Expected Output

public class CardTester{
  public static void main(String [] args){
  System.out.println("1-----------------");
  MFS bkash = new MFS("12121212");
  System.out.println("2-----------------");
  Card visaCard = new Card(2000, "VISA");
  System.out.println("3-----------------");
  bkash.addMoneyFromCard(5000);
  System.out.println("4-----------------");
  visaCard.addMoneyFromAccount(500, bkash);
  System.out.println("5-----------------");
  bkash.addNewCard(visaCard);
  System.out.println("6-----------------");
  bkash.addMoneyFromCard(5000);
  System.out.println("7-----------------");
  bkash.addMoneyFromCard(800);
  System.out.println("8-----------------");
  visaCard.addMoneyFromAccount(500, bkash);
  }
}
	1-----------------
	Account created.
	AccNo: 12121212
	Balance: 0.0 tk
	Card not saved.
	2-----------------
	VISA Card created successfully.
	Balance: 2000.0 tk
	3-----------------
	Please add a card to your MFS account
	4-----------------
	Insufficient account balance!
	5-----------------
	VISA Card added to MFS acc#12121212
	6-----------------
	Insufficient balance in VISA Card.
	7-----------------
	Transfer successful.
	New MFS account balance: 800.0 tk.
	Card balance: 1200.0 tk
	8-----------------
	Added 500 tk to card from MFS account#12121212.
	New balance in card: 1700.0 tk

Task 2

Design the **necessary** classes for the NetFlix system with the necessary properties to produce the given output for the provided driver code.

Driver code	Output

public class NetflixAccount{
  public static void main(String[] args){
  Movie m1 = new Movie("Avengers", "Sci-fi", "1:59:08");
  Movie m2 = new Movie("Lord of the Rings", "Fantasy", "2:58:05");
  Movie m3 = new Movie("Final Destination","Horror");
  Movie m4 = new Movie("Fast and Furious","Action", "2:05:04");
  System.out.println("==========");
  NetflixUser user1 = new NetflixUser("Martin Freeman", "Free");
  user1.addToFavourites(m1);
  user1.showFavourites();
  user1.upgradePlan();
  user1.addToFavourites(m1);
  user1.showFavourites();
  System.out.println("==========");
  NetflixUser user2 = new NetflixUser("Vin Diesel", "Premium");
  user2.addToFavourites(m1);
  user2.addToFavourites(m2);
  user2.addToFavourites(m3);
  user2.addToFavourites(m4);
  user2.showFavourites();
  System.out.println("==========");
  NetflixUser user3 = new NetflixUser("Chris Evans");
  user3.addToFavourites(m1);
  user3.upgradePlan();
  user3.addToFavourites(m2);
  user3.addToFavourites(m3);
  user1.upgradePlan();
  }
}
	==========
	New user account created!
	Cannot add movies to favourites.
	Please upgrade to Netflix Premium.
	No available favourites. Please upgrade to Netflix Premium.
	Welcome to Netflix Premium!
	Movie added to favourites.
	Showing Martin Freeman's Favourites:
	Avengers, Sci-fi, Duration: 1:59:08
	==========
	New user account created!
	Movie added to favourites.
	Movie added to favourites.
	Movie added to favourites.
	Cannot add new movies. Favourites list is full.
	Showing Vin Diesel's Favourites:
	Avengers, Sci-fi, Duration: 1:59:08
	Lord of the Rings, Fantasy, Duration: 2:58:05
	Final Destination, Horror, Duration: Unknown
	==========
	New user account created!
	Cannot add movies to favourites.
	Please upgrade to Netflix Premium.
	Welcome to Netflix Premium!
	Movie added to favourites.
	Movie added to favourites.
	You already have Netflix Premium!

Task 3

```
1  public class ExamClass {
2      public int ques;
3      public int sum;
4      public void methodA() {
5          System.out.println(ques + " " + 0 + " " + 0);
6      }
7  }
8  class QuizA {
9      public int x, y;
10     public int sum = 1;
11     public QuizA(int x, int y) {
12         this.x = y;
13         this.y = x;
14     }
15     public void methodA() {
16         int x = 3;
17         y = this.y + x;
18         ExamClass exam = new ExamClass();
19         exam.sum = x;
20         exam.ques = this.y;
21         x = this.x + x + exam.sum;
22         this.y = this.sum + methodB(exam.ques, exam);
23         System.out.println(x + " " + this.y + " " + sum);
24         sum = x % 2 + this.x;
25         y = x + y + exam.sum;
26         System.out.println(x + " " + y + " " + sum);
27     }
28     public int methodB(int x1, ExamClass x2) {
29         int y = 0;
30         y = this.y + x2.sum;
31         x2.ques = x1 + x2.ques;
32         sum = sum + x + y;
33         System.out.println(this.x + " " + this.y + " " + sum);
34         return x2.sum;
35     }
36 }
```

Driver Code	Output		

`public class QuizTesterA{`
`  public static void main(String []args){`
`  QuizA q1 = new QuizA(3,4);`
`  q1.methodA();`
`  }`
`}`			

HOMEWORK

Task 1

Design the Company and Employee classes so that the Tester1 class produces the given outputs.

Restriction: Company class can't have more than 1 array.

Driver Code	Output

public class Tester1{
  public static void main(String args[]){
  Employee e1 = new Employee();
  Employee e2 = new Employee("Alif", 34, "Fulltime");
  System.out.println("1-------------");
  Company c1 = new Company();
  c1.details();
  System.out.println("2-------------");
  Employee e3 = new Employee("Akter", 36,"Part-time");
  Employee e4 = new Employee("Ria", 38, "Fulltime");
  System.out.println("3-------------");
  c1.addEmployee(e2);
  c1.addEmployee(e3);
  System.out.println("4-------------");
  c1.details();
  System.out.println("5-------------");
  c1.addEmployee(e4);
  c1.addEmployee(e1);
  System.out.println("6-------------");
  c1.details();
  System.out.println("7-------------");
  c1.removeEmployee(e4);
  System.out.println("6-------------");
  c1.details();
  }
}
	A default employee has been created
	1-------------
	Company Name: ABC Company
	Total Employee: 0
	Fulltime Employees:
	Part-Time Employees:
	2-------------
	3-------------
	Alif has joined the company
	Akter has joined the company
	4-------------
	Company Name: ABC Company
	Total Employee: 2
	Fulltime Employees:
	Name: Alif, ID: 34
	Part-Time Employees:
	Name: Akter, ID: 36
	5-------------
	Ria has joined the company
	No more vacancy
	6-------------
	Company Name: ABC Company
	Total Employee: 3
	Fulltime Employees:
	Name: Alif, ID: 34
	Name: Ria, ID: 38
	Part-Time Employees:
	Name: Akter, ID: 36
	7-------------
	Ria has left the company
	6-------------
	Company Name: ABC Company
	Total Employee: 2
	Fulltime Employees:
	Name: Alif, ID: 34
	Part-Time Employees:
	Name: Akter, ID: 36

Task 2

Design the **Student** and **Department** class with the necessary properties so that the provided driver code generates the output given below. For simplicity, assume that a department can add a maximum of 5 students.

Driver Code	Output

public class DepartmentTester {
 public static void main(String[] args) {
  Student s1 = new Student("Akib", 10, 3.29);
  Student s2 = new Student("Reza", 15, 3.45);
  Student s3 = new Student("Kabir", 20,4.0);
  System.out.println("1==========");
  Department cse = new Department("CSE");
  cse.findStudent(-100);
  System.out.println("2==========");
  cse.addStudent(s1, s2, s3);
  System.out.println("3==========");
  cse.details();
  System.out.println("4==========");
  cse.findStudent(15);
  System.out.println("5==========");
  Student s4 = new Student("Nakib", 15,3.22);
  cse.addStudent(s4);
  System.out.println("6==========");
  s4.updateId(25);
  cse.addStudent(s4);
  System.out.println("7==========");
  cse.details();
  System.out.println("8==========");
  Student s5 = new Student("Sakib", 30,2.29);
  cse.addStudent(s5);
  System.out.println("9==========");
  cse.details();
 }
}
	1==========
	Student with this ID doesn't exist, Please give a valid ID
	2==========
	Welcome to CSE department, Akib
	Welcome to CSE department, Reza
	Welcome to CSE department, Kabir
	3==========
	Department Name: CSE
	Number of student:3
	Details of the students:
	Student name: Akib, ID: 10, cgpa: 3.29
	Student name: Reza, ID: 15, cgpa: 3.45
	Student name: Kabir, ID: 20, cgpa: 4.0
	4==========
	Student info:
	Student Name: Reza
	ID: 15
	CGPA: 3.45
	5==========
	Student with the same ID already exists, Please try with another ID
	6==========
	Welcome to CSE department, Nakib
	7==========
	Department Name: CSE
	Number of student:4
	Details of the students:
	Student name: Akib, ID: 10, cgpa: 3.29
	Student name: Reza, ID: 15, cgpa: 3.45
	Student name: Kabir, ID: 20, cgpa: 4.0
	Student name: Nakib, ID: 25, cgpa: 3.22
	8==========
	Welcome to CSE department, Sakib
	9==========
	Department Name: CSE
	Number of student:5
	Details of the students:
	Student name: Akib, ID: 10, cgpa: 3.29
	Student name: Reza, ID: 15, cgpa: 3.45
	Student name: Kabir, ID: 20, cgpa: 4.0
	Student name: Nakib, ID: 25, cgpa: 3.22
	Student name: Sakib, ID: 30, cgpa: 2.29

Task 3

Design the **necessary** classes for the UberEats system with the necessary properties to produce the given output for the provided driver code

Driver code	Output

public class UberEatsAccount{
  public static void main(String[] args){
  System.out.println("==========");
  UberEatsUser user1 = new UberEatsUser("Peter Parker", "Badda");
  UberEatsUser user2 = new UberEatsUser("Matt Murdock", "Mohammadpur");
  UberEatsUser user3 = new UberEatsUser("Reed Richards");
  UberEatsUser user4 = new UberEatsUser("Peggy Carter", "Mirpur");
  Restaurant r1 = new Restaurant("Chillox", "Badda");
  r1.takeOrder(user1);
  r1.takeOrder(user2);
  r1.takeOrder(user3);
  r1.takeOrder(user4);
  r1.completeOrders();
  System.out.println("==========");
  Restaurant r2 = new Restaurant("Kyoshi", "Gulshan");
  r2.takeOrder(user3);
  user3.updateLocation("Malibagh");
  user3.updateLocation("Bashundhara");
  r2.takeOrder(user3);
  r2.takeOrder(user3);
  r2.completeOrders();
  r2.completeOrders();
  System.out.println("==========");
  Restaurant r3 = new Restaurant("Cilantro","Banani");
  r3.takeOrder(user1);
  r3.takeOrder(user2);
  r3.takeOrder(user3);
  r3.takeOrder(user4);
  }
}
	==========
	Your order has been added!
	Your order has been added!
	Location : Unknown. Please update your location information!
	Your order has been added!
	Showing Chillox's orders:
	Order by Peter Parker at Badda completed
	Order by Matt Murdock at Mohammadpur completed
	Order by Peggy Carter at Mirpur completed
	==========
	Location : Unknown. Please update your location information!
	Update Successful!
	We already have your location.
	Please place an order!
	Your order has been added!
	You already have a pending order!
	Showing Kyoshi's orders:
	Order by Reed Richards at Malibagh completed
	No pending orders at the moment.
	==========
	Your order has been added!
	Your order has been added!
	Your order has been added!
	We are really busy right now.
	Please order after some time.
	Thank you!

Task 4

Design the **ConnectFriends** class with the necessary properties so that the provided driver code generates the output given below.

Driver Code	Output

public class ConnectTester{
  public static void main(String[] args) {
  ConnectFriends sanaf = new ConnectFriends("Sanaf");
  System.out.println("=======1=======");
  ConnectFriends mantasha = new ConnectFriends("Mantasha", 3);
  ConnectFriends mostafiz = new ConnectFriends("Mostafiz");
  ConnectFriends matt = new ConnectFriends("Matt", 4);
  System.out.println("=======2=======");
  sanaf.sendFriendRequest(mantasha);
  System.out.println("=======3=======");
  sanaf.sendFriendRequest(mostafiz, matt);
  System.out.println("=======4=======");
  sanaf.showDetails();
  System.out.println("=======5=======");
  sanaf.removeRequest("Mantasha");
  System.out.println("=======6=======");
  sanaf.showDetails();
  System.out.println("=======7=======");
  sanaf.removeRequest("Murdock");
  System.out.println("=======8=======");
  sanaf.removeRequest("Matt");
  sanaf.removeRequest("Mostafiz");
  sanaf.showDetails();
  System.out.println("=======9=======");
  mantasha.showDetails();
  }
}
	Welcome to ConnectFriends, Sanaf
	=======1=======
	Welcome to ConnectFriends, Mantasha
	Welcome to ConnectFriends, Mostafiz
	Welcome to ConnectFriends, Matt
	=======2=======
	Sanaf sent a friend request to Mantasha.
	=======3=======
	Sanaf sent a friend request to Mostafiz.
	Sanaf has reached the friend request limit!
	=======4=======
	User Name: Sanaf
	Maximum number of Sent Friend Request: 2
	Total Friends Request: 2
	Sent Friends Request: Mantasha Mostafiz
	=======5=======
	Reuqest to add Mantasha is removed for Sanaf.
	=======6=======
	User Name: Sanaf
	Maximum number of Sent Friend Request: 2
	Total Friends Request: 1
	Sent Friends Request: Mostafiz
	=======7=======
	Murdock is not in Sanaf's sent request list.
	=======8=======
	Matt is not in Sanaf's sent request list.
	Reuqest to add Mostafiz is removed for Sanaf.
	User Name: Sanaf
	Maximum number of Sent Friend Request: 2
	Total Friends Request: 0
	Sent Friends Request:
	=======9=======
	User Name: Mantasha
	Maximum number of Sent Friend Request: 3
	Total Friends Request: 0
	Sent Friends Request:

Task 5

```
1  public class Trace {
2   public int sum, temp;
3   public Trace(int sum, int temp){
4     this.sum = sum;
5     this.temp = temp;
6   }
7  }
8  class Quiz5{
9   public int sum = 11, x = -2, y = 16;
10  public Trace trace = null;
11  public Quiz5(Trace t){
12    trace = t;
13    int x = trace.temp + y;
14    sum = sum + (t.sum++) + y;
15    System.out.println(trace.sum + " " + sum + " " + x);
16    sum -= 10;
17  }
18  public void methodA(int sum, int temp){
19    sum = -3 + this.sum - trace.sum;
20    x = sum + 13 + y;
21    y = trace.temp + temp + sum;
22    this.sum = y + methodB(trace.temp, trace) + trace.temp;
23    System.out.println(sum + " " + y + " " + this.sum);
24  }
25  public int methodB(int x, Trace temp){
26    int sum = x + temp.sum + this.x;
27    temp.temp = sum + this.sum;
28    System.out.println(x + " " + temp.temp + " " + sum);
29    return sum;
30  }
31 }
```

```
Trace p = new Trace(3, 4);
Quiz5 q = new Quiz5(p);
q.methodA(4, 8);
q.methodA(5, 10);
```

Output		

Task 6

```
1  public class Foo{
2    public int bar, buz;
3    public Foo(int bar, int buz){
4      this.bar = bar;
5      this.buz = buz;
6    }
7  }
8  class Quiz5{
9    public int sum = 12, x = 2, y = 6;
10   public Foo foo;
11   public Quiz5(Foo f){
12     foo = f;
13     int x = this.foo.buz + y;
14     sum = sum + (f.bar--) + y;
15     System.out.println(foo.bar + " " + sum + " " + x);
16     sum -= 10;
17   }
18   public void methodA(int bar, int buz){
19     bar = 3 + bar - this.foo.bar;
20     x = bar + 12 + y;
21     y = foo.buz + buz + bar;
22     sum = y + methodB(foo.buz, foo) + foo.buz;
23     System.out.println(bar + " " + y + " " + sum);
24   }
25   public int methodB(int bar, Foo buz){
26     int sum = bar + buz.bar + x;
27     buz.buz = sum + this.sum;
28     System.out.println(bar + " " + buz.buz + " " + sum);
29     return sum;
30   }
31 }
```

Driver Code	Output		

`public class LabTester{`
`  public static void main(String []args){`
`  Foo p = new Foo(13, 4);`
`  Quiz5 q = new Quiz5(p);`
`  q.methodA(14, 8);`
`  }`
`}`			