CSC 110 2.0 Object Oriented Programming

Tutorial 06

Instructions:

- All questions must be attempted and answers submitted in a handwritten document, **on or before, 4.00pm on Monday, 26th August 2019, to the Department Office**.
- You must indicate your **Index Number and the Tutorial Class** to which you belong to **(LCS1/ LCS2/ NFC3.1)** clearly on the front page of your submission.

A. Understanding Code Segments

1. **For each of the code segments below, write the output of the main method in the respective Driver Class.**

(i) Shapes

```
abstract class Shape{
     abstract void draw();
}

class Rectangle extends Shape{
     void draw(){
     System.out.println("drawing rectangle");
     }
}
class Circle extends Shape{
     void draw(){
     System.out.println("drawing circle");
     }
}

class TestAbstraction{
     public static void main(String args[]){
     Shape s=new Circle();
     s.draw();
     }
}
```

(ii) Banks

```
abstract class Bank{
     abstract int getRateOfInterest();
}
class SBI extends Bank{
     int getRateOfInterest(){
          return 7;
     }
}
class PNB extends Bank{
     int getRateOfInterest(){
          return 8;
     }
}
```

```
class TestBank{
     public static void main(String args[]){
     Bank b;
     b=new SBI();
     System.out.println("Rate of Interest is: "+b.getRateOfInterest()+" %");
     b=new PNB();
     System.out.println("Rate of Interest is: "+b.getRateOfInterest()+" %");
}}
```

(iii) Person with a Job

```
public class Job {
    private String role;
    private long salary;
    private int id;
    public String getRole() {
        return role;
    }
    public void setRole(String role) {
        this.role = role;
    }
    public long getSalary() {
        return salary;
    }
    public void setSalary(long salary) {
        this.salary = salary;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
}

public class Person {
    private Job job;
    public Person(){
        this.job=new Job();
        job.setSalary(1000L);
    }
    public long getSalary() {
        return job.getSalary();
    }
}

public class TestPerson {
    public static void main(String[] args) {
        Person person = new Person();
        long salary = person.getSalary();
    }
}
```

2. **Based on the following Classes, write the necessary code segments within the main method to get the shown output.**

```
class Product {
    private int id;
    private String name;
    private String description;
    public Product(int id, String name, String description) {
         super();
         this.id = id;
         this.name = name;
         this.description = description;
    }
    public int getId() {
         return id;
    }
    public void setId(int id) {
         this.id = id;
    }
    public String getName() {
         return name;
    }
    public void setName(String name) {
         this.name = name;
    }
    public String getDescription() {
         return description;
    }
    public void setDescription(String description) {
         this.description = description;
    }
    @Override
    public String toString() {
         return "Product [id=" + id + ", name=" + name + ", description=" + description
+ "]";
    }
}
```

```
class LineItem {
    private int id;
    private int quantity;
    private Product p;
    public LineItem(int id, int quantity, Product p) {
         super();
         this.id = id;
         this.quantity = quantity;
         this.p = p;
    }
    public int getId() {
         return id;
    }
    public void setId(int id) {
         this.id = id;
    }
```

```java
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
    public Product getP() {
        return p;
    }
    public void setP(Product p) {
        this.p = p;
    }
    @Override
    public String toString() {
        return "LineItem [id=" + id + ", quantity=" + quantity + ", p=" + p + "]";
    }
}
```

```java
class Order {
    private int id;
    private String name;
    private List<LineItem> lineItems;
    public Order(int id, String name) {
        super();
        this.id = id;
        this.name = name;
        this.lineItems = new ArrayList<LineItem>();
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    @Override
    public String toString() {
        return "Order [id=" + id + ", name=" + name + ", lineItems=" + lineItems + "]";
    }
    public void addItem(int id, int quantity, Product p) {
        this.lineItems.add(new LineItem(id, quantity, p));
    }
}
```

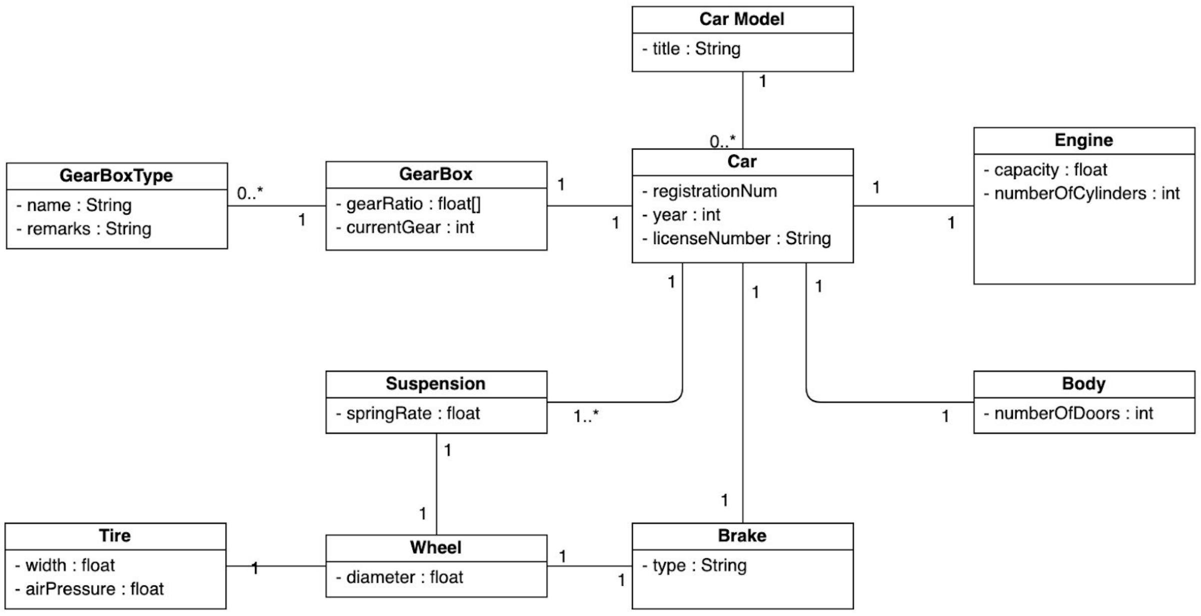
The output required is shown below:

```
Order ---
Order [id=1, name=ORD#1, lineItems=[LineItem [id=1, quantity=2, p=Product [id=1,
name=Pen, description=This is red pen]], LineItem [id=2, quantity=1, p=Product [id=2,
name=Pencil, description=This is pencil]], LineItem [id=3, quantity=5, p=Product [id=3,
name=ColorBox, description=This is color box]]]]
```

You are required to write a Driver Class called “TestOrder” with a main method that can generated the above-mentioned output.

B. Coding in Java

For the following UML class diagram of a system to store details, write the source code in Java following Object Oriented Programming concepts.



Note:
You are only required to implement all Constructors, Getters and Setters for the Car, Suspension and Wheel classes.

C. Solving Problems: the Object-Oriented way

Read the given passage and design a solution to the said problem using Object Oriented Programing Concepts.

A hockey league is made up of at least four hockey teams. Each hockey team is composed of six to twelve players, and one player captains the team. A team has a name and a record. Players have a number and a position. Hockey teams play games against each other. Each game has a score and a location. Teams are sometimes lead by a coach. A coach has a level of accreditation and a number of years of experience, and can coach multiple teams. Coaches and players are people, and people have names and addresses. Addresses include house number, street name, town and postal code.

* * * * * * * *