King Saud University

College of Computer and Information Sciences

Computer Science Department

Course Code:	CSC 113
Course Title:	Computer Programming II
Semester:	Winter 2022
Exercises Cover Sheet:	**Midterm Exam**

Student Name:	
Student ID:	
Student Section No/ Time.	

Tick the Relevant	**Computer Science B.Sc. Program ABET Student Outcomes**	**Question No. Relevant Is Hyperlinked**	**Covering%**
	a) Apply knowledge of computing and mathematics appropriate to the computer science;		
	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution		
	c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;		
	d) Function effectively on teams to accomplish a common goal;		
	e) Understanding of professional, ethical, legal, security, and social issues and responsibilities;		
	f) Communicate effectively with a range of audiences;		
	g) Analyze the local and global impact of computing on individuals, organizations and society;		
	h) Recognition of the need for, and an ability to engage in, continuing professional development;		
	i) Use current techniques, skills, and tools necessary for computing practices.		
	j) Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;		
	k) Apply design and development principles in the construction of software systems of varying complexity;		

Name :____________________________________ **ID:**___________________

QUESTION 1:

Write the answers in the following table: 10 marks

1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

1- What will be the output of the following Java program (assume no compilation/syntax errors)?

```
public class ObjA {
 public ObjB b;
    public ObjA(ObjB b, int v){
        this.b = b;
        b.y = v;
    }
}
```

```
public class ObjB {
    public int y;
    public ObjB(int y){
        this.y = y;
    }
}
```

```
    public static void main(String args[]) {
        ObjB b = new ObjB(5);
        ObjA a = new ObjA(b,4);
        System.out.print(a.b.y +",");
        b.y = 3;
        System.out.print(a.b.y);
    }
```

A. 5,3
B. 4,3
C. 5,5
D. 4,4

2- Which of the following statements about inheritance in Java is NOT true?
A. A superclass can have multiple subclasses.
B. A subclass can override inherited methods from its superclass.
C. Inheritance creates a "is-a" relationship between a subclass and its superclass.
D. A class can extend multiple classes.

3- Which of the following is NOT a reason to use inheritance in Java?
A. To reuse code from a superclass in a subclass.
B. To create a subclass that is a specific type of a superclass.
C. To create a subclass that can access the private fields of its superclass.
D. To create a subclass that has a unique behavior from its superclass.

4- What will be the output of the following Java program?

```
public class Animal {
    public void eat() {
        System.out.print("Animal
eating,");
    }
}
```

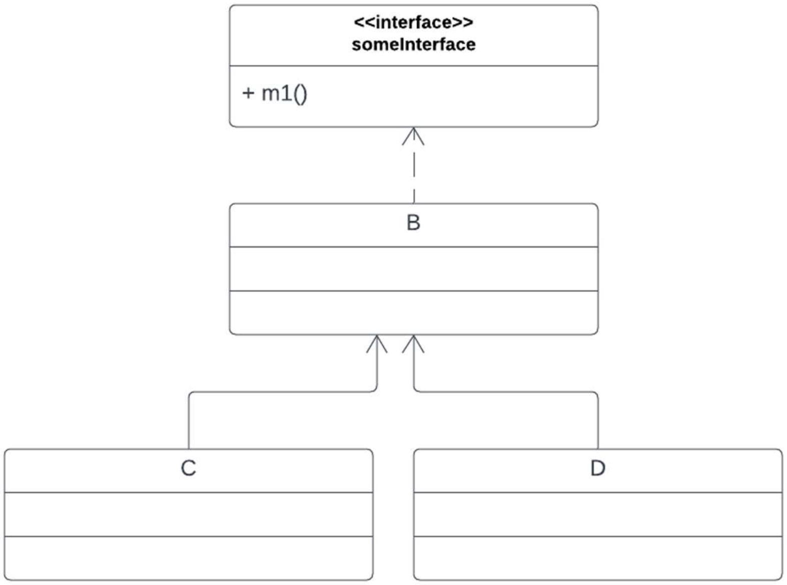
```
public class Cat extends Animal {
    public void eat() {
        System.out.print("Cat eating,");
    }
    public void meow() {
        System.out.print("Meow,");
    }
}
```

```
    public static void main(String args[]) {
        Animal a = new Cat();
        a.eat();
        ((Cat) a).meow();     }
}
```

A. Animal eating,Meow,
B. Cat eating,Meow,
C. Animal eating,Cat eating,
D. Error

5- Which of the following statement is NOT correct.
   A. Abstract method must be a part of an abstract class.
   B. Abstract class must contain an abstract method.
   C. Abstract class can’t be extended.
   D. Can’t create an instance of an abstract class

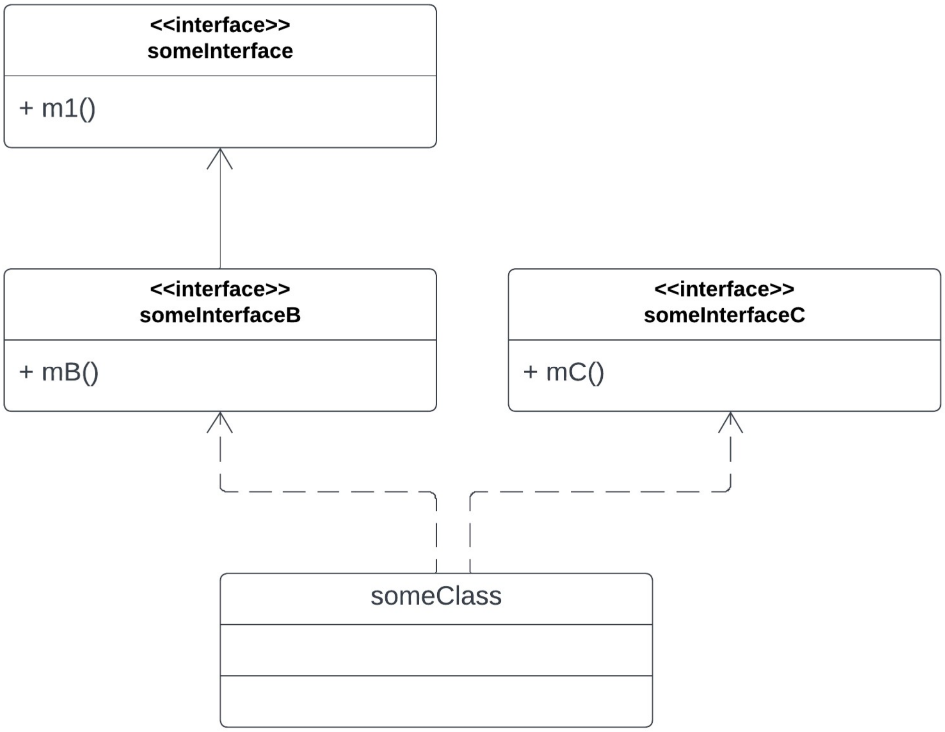
6- The option(s) to successfully compile the following class(es) as Concrete class(es):



A. Class B must implement m1().
B. Classes C and D must implement m1().
C. Either B, C, or D implements m1().
D. A or B

Name :____________________________________ ID:___________________

7- The option(s) to successfully compile the following class(es) as Concrete class(es):



A. `someClass` must implement `mB()`, `mC()`, and `m1()`.
B. `someClass` must implement `mB(), mC()`.
C. `someClass` can impmlement any of `mB(), mC(), or m1()`.
D. Error

8- Suppose we have a method called `m1()` in subclass, how can we call it's from super class?

A. this.m1();
B. m1();
C. super.m1();
D. None of the above

9- Which of the following statements is true about recursion and loops in Java in terms of memory usage?

A. Recursion is more memory-efficient than loops because it does not require the use of additional variables to track the loop counter.
B. Loops are more memory-efficient than recursion because they do not require the use of the call stack to store intermediate results.
C. Recursion and loops are equally memory-efficient, but loops are more readable and easier to understand.
D. Recursion and loops are equally memory-efficient, but recursion is more readable and easier to understand.

10- What will be the output of the following Java program (assume no compilation/syntax errors)?

```
    public static int m1(int a[], int low) {
        if (low == a.length-1) {
            return a[low-1];
        } else {
            int x = m1(a, low+1);
            if(a[low] > x)
                return a[low];
            else
                return x;
        }
}
```

```
public static void main(String
args[]) {
        int a[] = {3,5,1,7,8};

System.out.println(m1(a,0));
    }
```

A. 3
B. 5
C. 7
D. 8

Name :__________________________________ ID:___________________

Question 2: What is the output of main method (assume no compilation/syntax errors)?

```
public class Vehicle {
     protected String brand;  protected String madeIn; protected int wheelD;

     public Vehicle() {
          this("No Name","Germany",600);
     }

     public Vehicle(String b, String m, int w){
          brand = b; madeIn = m; wheelD = w;
     }

     public void display() {
          System.out.println("Brand: " + brand);
          System.out.println("Madein : " + madeIn);
          System.out.println("Wheel diameter: " + wheelD);
     }

     public void travel (int nbKm){
          display();
     } } // End Class Vehicle
--------------------------------------------------------------------------------
public class Car extends Vehicle{
     int fuel;  int consumption;

     public Car(){}

     public Car(String b, String m, int w, int f, int c){
          super(b,m,w);
          fuel = f;   consumption = c;
     }

     public void display(){
          super.display();
          System.out.println("Fuel: " + fuel);
          System.out.println("Consumption : " + consumption);
     }

     public void travel(int nbKm){
          fuel = calculate(nbKm);
          display();
     }

     public int calculate(int nbKm){
          if (nbKm <= 1)
               return 5;
          else
               return fuel - calculate(consumption * (nbKm/100));}} // End Class Car
--------------------------------------------------------------------------------
```

Name :__________________________________ **ID:**____________________

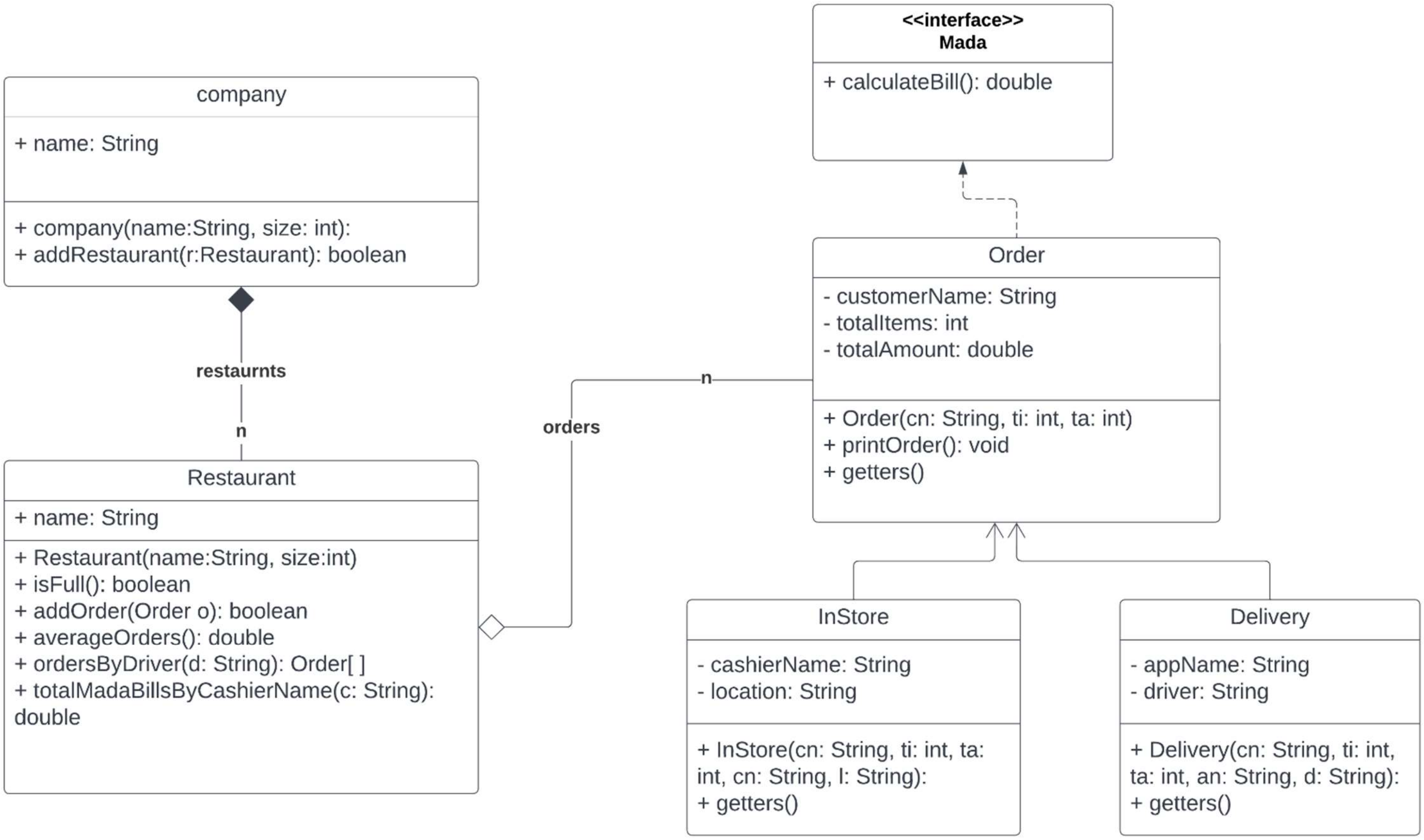
```
public static void main(String args[]) {
    Car m1 = new Car();
    Car m2 = new Car("Yamaha", "Japan", 700, 35, 4);
    m1.display();
    m2.travel(500);


}
```

Answer: 5 marks. 0.5/line

```
Brand: No Name
Madein : Germany
Wheel diameter: 600
Fuel: 0
Consumption : 0
Brand: Yamaha
Madein : Japan
Wheel diameter: 700
Fuel: 5
Consumption : 4
```

Name :__________________________________ **ID:**__________________

Question 3: ***Consider the following UML class diagram:*** ***10 marks***



<<interface>>
Mada
+ calculateBill(): double
company
+ name: String
+ company(name:String, size: int):
+ addRestaurant(r:Restaurant): boolean
restaurnts
n
Restaurant
+ name: String
+ Restaurant(name:String, size:int)
+ isFull(): boolean
+ addOrder(Order o): boolean
+ averageOrders(): double
+ ordersByDriver(d: String): Order[ ]
+ totalMadaBillsByCashierName(c: String): double
orders
n
Order
- customerName: String
- totalItems: int
- totalAmount: double
+ Order(cn: String, ti: int, ta: int)
+ printOrder(): void
+ getters()
InStore
- cashierName: String
- location: String
+ InStore(cn: String, ti: int, ta: int, cn: String, l: String):
+ getters()
Delivery
- appName: String
- driver: String
+ Delivery(cn: String, ti: int, ta: int, an: String, d: String):
+ getters()

Name :___________________________________ ID:___________________

Interface Class ***Mada:***

- Methods:
  - ***calculateBill():***this method calculates and return the bill amount of any order as follows:
    - totalAmount * 0.01

Class ***Order:***

- Attributes:
  - ***customerName*: ** name of the customer who placed the order.
  - ***totalItems:*** number of ordered items in the order.
  - ***totalAmount:*** total price of all items in the order.

- Methods:
  - ***Order(cn: String, ti: int, ta: int): constructor***.
  - ***printOrder():*** prints only the customer name and total price of the order.
  - ***getters() :*** return the value of each attribute.

Class ***InStore:***

- Attributes:
  - ***cashierName:*** the name of the cashier.
  - ***location***: location of the store
- Methods:
  - ***InStore(cn: String, ti: int, ta: int, cn: String, l: String):constructor.***
  - ***getters() :*** return the value of each attribute.

Class ***Delivery:***

- Attributes:
  - ***appName:*** the name of the application used to place the order.
  - ***driver:*** the name of driver who will deliver the order.
- Methods:
  - ***Delivery(cn: String, ti: int, ta: int, an: String, d: String): constructor***.
  - ***getters() :*** return the value of each attribute.

Class ***Restaurant:***

- Attributes:
  - ***name:*** the name of the Restaurant.
- Methods:
  - ***Restaurant (name:String, size:int): constructor***

    - ***isFull():*** return true if the restaurant is full, and false otherwise.
    - ***addOrder(o: Order):*** This method will add the order ***o*** to the array. The method will return true if the order is added successfully. Otherwise, it will return false.
    - ***averageOrders ():*** This method will return the average of all orders in the restaurant.
    - **ordersByDriver (d: String):** This method will return array of order delivered by driver ***d***.
    - ***totalMadaBillsByCashierName(c: String):*** this method will return the total amount of Mada bills by cashier ***c***.

Class ***Company***

- Attributes:
    - ***name:*** the name of the company.
- Methods:
    - ***Company*** **(*name:String, size: int*).** ***constructor***
    - **addRestaurant (r: Restaurant):** This method will the restaurant **r** to the array.

Name :__________________________________ **ID:**____________________

1- **Answer the following questions:**

A. Name the class(es) that will implement (have code) `calculateBill()` method?

Answer: order **0.5 mark**

B. Name the class(es) that will implement (have code) `printOrder()` method?

Answer: order **0.5 mark**

2- **Complete the following methods.**

```
public double averageOrders(){     3 marks

          double sum = 0;

          for(int i=0; i<nb;i++)

           sum+= orders[i].getTotalAmount();

     return sum/nb;

}

public Order[] ordersByDriver (String d) {  3 marks


     Order driver[] = new Order[nb]; int count =0;

          for(int i=0; i<nb;i++)

               if(orders[i] instanceof Delivery &&
(Delivery)orders[i].getDriverName.equals(d))

     driver[count++] = orders[i];


return driver;
}

public double  totalMadaBillsByCashierName (String c) {3 marks

          double sum = 0;
          for(int i=0; i<nb;i++)
           if(orders[i] instanceof InStore &&
(InStore)orders[i].getChshierName.equals(c))
          sum+= orders[i].calcualteBill();
     return sum;

}
```