



SECJ2154
OBJECT-ORIENTED PROGRAMMING

EXERCISE: ABSTRACT

SECTION 02

LECTURER NAME:
DR. NUR EILYAH

NAME:
MADINA SURAYA BINTI ZHARIN

MATRIC NUMBER:
A20EC0203

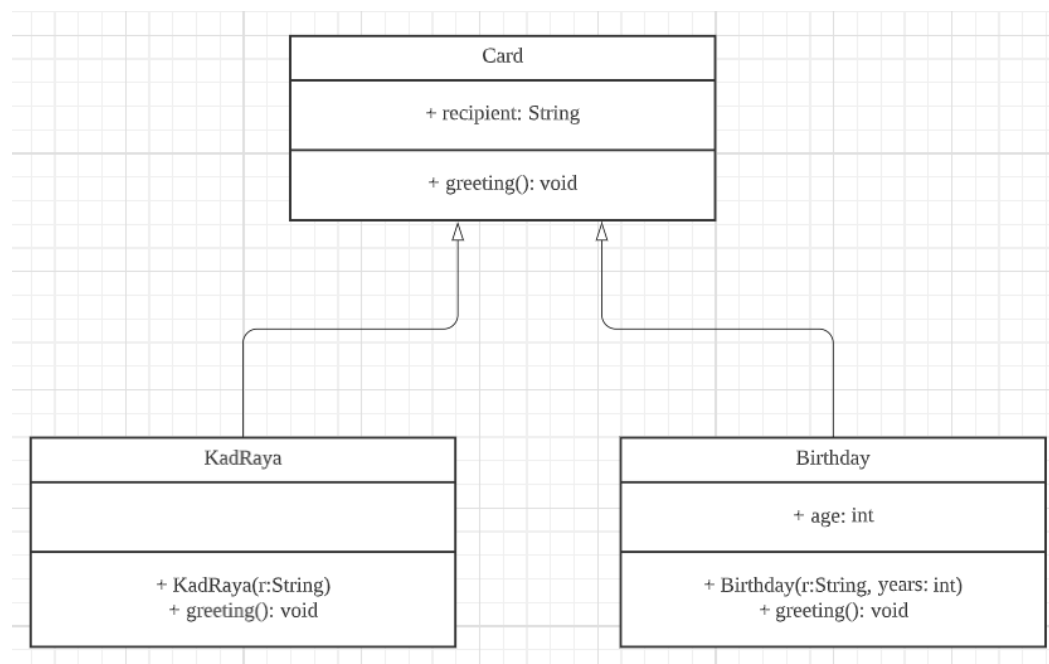
Answers:

Question 1

1. Illegal
2. Illegal
3. Legal
4. Legal
5. Legal

Question 2

i.



ii.

```
class CardTester {
    public static void main (String[] args) {
        KadRaya a = new KadRaya("Madina");
        Birthday b = new Birthday("Madina",22);
        a. greeting();
        b. greeting();
    }
}
```

iii.

```
class Wedding extends Card {
    int number;
    public Wedding (String r, int years) {
        recipient = r;
        number = years;
    }

    System.out.println("Dear" + recipient + ", \n");
    System.out.println("Happy" + number + "th Anniversary \n");
}

class CardTester {
    public static void main (String[] args) {
        KadRaya a = new KadRaya("Madina");
        Birthday b = new Birthday("Madina",22);
        Wedding w = new Wedding("Madina",1);
        a. greeting();
        b. greeting();
        w. greeting();
    }
}
```

iv.

```
public abstract class Card {
    protected String recipient;
    public abstract void greeting();
}
```

v.

```
public abstract class Card {
    private String recipient;

    public Card (String recipient){
```

```

        this.recipient = recipient;
    }
    public String getRecipient() {
        return recipient;
    }
    public void setRecipient(String recipient) {
        This.recipient = recipient;
    }
    public abstract void greeting();
}

```

```

class Birthday extends Card {
    int age;
    public Birthday (String r, int years) {
        super(r);
        age = years;
    }

    public void greeting() {
        System.out.println("Dear" + getRecipient() + ", \n");
        System.out.println("Happy" + age+ "th Birthday \n\n");
    }
}

```

```

class Wedding extends Card {
    int number;

    public Wedding (String r, int years) {
        super(r);
        number = years;
    }

    public void greeting() {

```

```
        System.out.println("Dear" + getRecipient() + ", \n");
        System.out.println("Happy" + number + "th Anniversary \n\n");
    }}

class CardTester {
    public static void main (String[] args) {
        KadRaya a = new KadRaya("Madina");
        Birthday b = new Birthday("Madina",22);
        Wedding w = new Wedding("Madina",1);

        a. greeting();
        b. greeting();
        w. greeting();
    }}
}
```

Question 3

```
public class Hexagon extends GeometricObject implements Comparable {  
    private double side;  
    public Hexagon(double side){  
        this.side = side;  
    }  
    public double findArea() {  
        return 3 * Math.sqrt(3) * side * side;  
    }  
    public double findPerimeter() {  
        return side * 6;  
    }  
    public int compareTo(Object obj) {  
        Hexagon rhs = (Hexagon)obj;  
        if(side > rhs.side)  
            return 1;  
        else if(side < rhs.side)  
            return -1;  
        else  
            return 0;  
    }  
}
```