Task 2

Subject	Data Structure and Algorithm
Lecturer	Imam Fahrur Rozi ST. MT.
Type	Assignment
Semester	Semester 2
Time	@February 23, 2023
Files & Media	

Question

1. It has Attributes and Method
2. public class (className)
3. 4, namaBarang: String, jenisBarang: String, stok: int, hargaSatuan: int
4. 4, tampilBarang(): void, tambahStok(n: int): void, kurangiStok(n: int): void, hitungHargaTotal(jumlah: int): int
5. 4, tampilBarang(), tambahStok(int n), kurangiStok(int n), hitungHargaTotal(int jumlah)
6. line 7, 13, 16, 19
7.

```
void kurangStok(int n)
{
  stok <= 0 ? return : stok = stok-n;
}
```

8. because stok is int, and if we use other than int it will change the paramater too
9. because we want to return the value of jumlah by multiplying it with hargaSatuan
10. because it need the stok
11. line 5, and the name of resulting object is b1
12. we can use dot notation (.) in the instance of the class
13. in the line 9 until 14
14. it used to print the name, type, stock, and the price of the barang

15.

```
Barang b3 = new Barang ("AMG GT-Four", "Car", 3_460_000_000, 69);
```

Task

1. Code

```
public class Lingkaran
{
  double phi = Math.phi, double r;

  public void double hitungLuas(int r)
  {
    return phi*(r*r);
  }

  public void double hitungKeliling(int r)
  {
    return phi*(r*2);
  }

  public static void Main (String[] args)
  {
    System.out.println(hitungLuas(7));
    System.out.println(hitungKeliling(7));
}
```

2. Diagram Class and Code

RentalTransaction
memberID: String memberName: String gameName: String dailyPrice: int dayRent: int
printRental(): String rentalPrice(): int

```
public class RentalTransaction
{
    String memberID, memberName, gameName;
    int dailyPrice, dayRent;

    RentalTransaction(String mid, String mn, String gn, int dp, int dr)
    {
        memberID = mid;
        memberName = mn;
        gameName = gn;
        dailyPrice = dp;
        dayRent = dr;
    }

    int rentalPrice()
    {
        return dailyPrice*dayRent;
    }
    void printRental()
    {
        System.out.println("Member ID: " + memberID);
        System.out.println("Member Name: " + memberName);
        System.out.println("Game Name: " + gameName);
        System.out.println("Game Price: " + dailyPrice);
        System.out.println("Rent Duration: " + dayRent);
        System.out.println("Total Price: " + rentalPrice());
    }
}
```

```
public class RentalTransactionMain
{
    public static void main (String[] args)
    {
        RentalTransaction g1 = new RentalTransaction("69420", "bambang", "Crash Ngebambangcoot", 5_000, 7);
        g1.printRental();
    }
}
```

3. Code

```
public class Item
{
    String name;
    int unitPrice, qty;

    Item(String nm, int up, int qt)
    {
        name = nm;
        unitPrice = up;
        qty = qt;
    }

    int calculateTotalPrice()
    {
        return unitPrice*qty;
    }

    int calculateDisc()
    {
        if (unitPrice*qty > 100_000)
        {
            unitPrice = (int) (unitPrice * 0.1);
        }
        else if (unitPrice*qty >= 50_000 && unitPrice*qty <= 100_000)
        {
            unitPrice = (int) (unitPrice * 0.05);
        }
        return unitPrice;
    }

    int calculateFinalPrice()
    {
        unitPrice = calculateTotalPrice() - (calculateDisc());
        return unitPrice;
    }

    public void printFinalPrice()
    {
        System.out.println("Name = " + name);
        System.out.println("Price = " + unitPrice);
        System.out.println("Quantity = " + qty);
        System.out.println("Total Price = " + calculateTotalPrice());
        System.out.println("Discount = " + calculateDisc());
        System.out.println("Final Price = " + calculateFinalPrice());
    }
}
```

```
public class ItemMain
{
    public static void main (String[] args)
    {
        Item i1 = new Item ("Baju", 69_000, 1);
        i1.printFinalPrice();
    }
}
```

4. Code

```
public class Pacman
{
    public int x;
    public int y;
    public int width;
    public int height;

    public int moveLeft()
    {
        int move = x - 1;
        if (move >= 0)
        {
            x = move;
        }
        return x;
    }

    public int moveRight()
    {
        int move = x + 1;
        if (move <= width)
        {
            x = move;
        }
        return x;
    }

    public int moveUp()
    {
        int move = y + 1;
        if (move <= height)
        {
            y = move;
        }
        return y;
    }

    public int moveDown()
    {
        int move = y - 1;
        if (move >= 0)
        {
            y = move;
        }
        return y;
    }

    public void printPosition()
    {
        System.out.println("Position x: " + x);
        System.out.println("Position y: " + y);
    }
}
```