Object Oriented Programming Job Sheet 2

From:

AL AZHAR RIZQI RIFA'I FIRDAUS

Class:

2 I

Absence:

01

Student Number Identity:

2241720263

Department:

Information Technology

Study Program:

Informatics Engineering

Experiment 1: Create Diagram Class

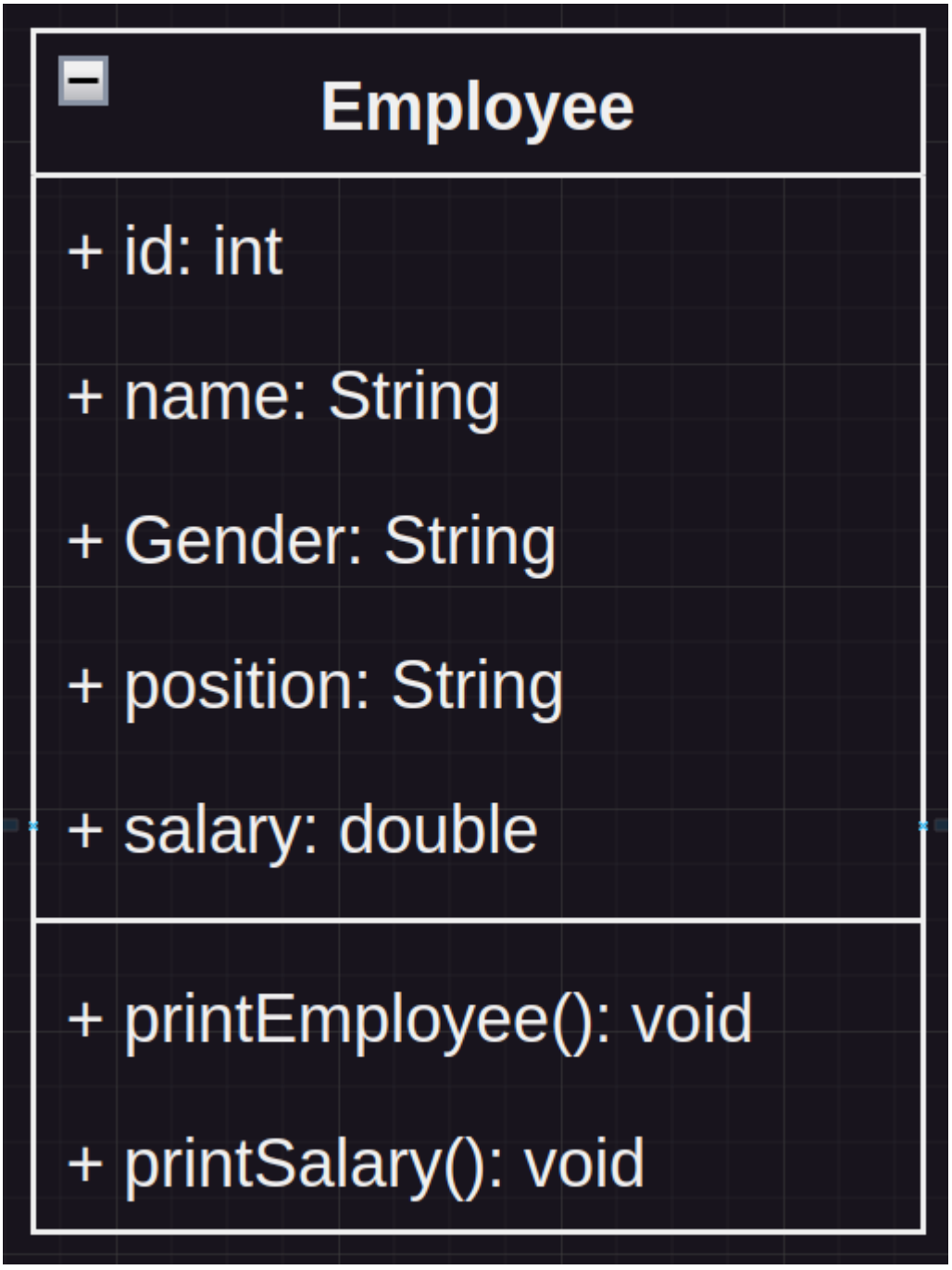
Case Study 1:

In a company, one of the data processed is employee data. Each

an employee has an id, name, gender, job title, position, and salary. Each employee

can also display their personal data and see their salary.

1. Draw the class diagram design of case study 1!



2. List the classes that can be created from case study 1!

- **Employee**

3. Mention the attributes and their data types that can be identified from each class from case study 1!

- **id: int**
- **name: String**
- **Gender: String**
- **position: String**
- **salary: double**

4. Mention the methods that you have created from each class in case study 1!

case study 1!

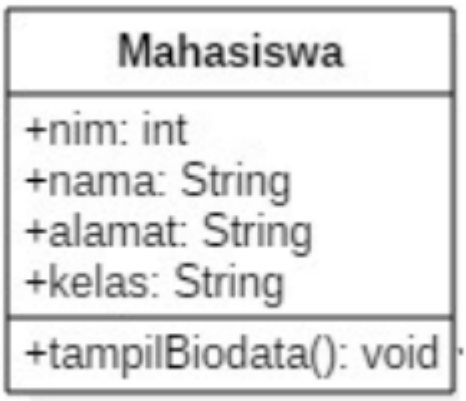
- **printEmployee(void)**
- **printSalary(void)**

Experiment 2: Creating and accessing members of a class

Case Study 2:

Look at the class diagram below. Create a program based on the class diagram

class diagram!



Work steps:

1. Open a text editor or IDE, e.g. Notepad++ / netbeans.

2. Type the following program code:

Students.java × TestStudents.java

src > Exp2 > Students.java > Students > printStudents()

```java
package Exp2;

public class Students {
    public int nim;
    public String name;
    public String address;
    public String class_;

    Codeium: Refactor | Explain | Generate Javadoc
    public void printStudents() {
        System.out.println("Nim : " + nim);
        System.out.println("Name : " + name);
        System.out.println("Address : " + address);
        System.out.println("Class : " + class_);
    }
}
```

3. Save it with the file name Student.java.

4. To be able to access the members of an object, an instance must be created first.

of the class first. The following is how to access the members of the

members of the Student class by opening a new file then typing the following program code

the following program code:

Students.java | TestStudents.java

src > Exp2 > TestStudents.java > TestStudents > main(String[])

```java
package Exp2;

public class TestStudents {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Students students1 = new Students();
        students1.nim = 123;
        students1.name = "Azhar";
        students1.address = "Soekarno Hatta Street";
        students1.class_ = "2I";
        students1.printStudents();
    }
}
```

5. Save the file with TestMahasiswa.java

6. Run the TestStudent class

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
ceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-2/code
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Nim : 123
Name : Azhar
Address : Soekarno Hatta Street
Class : 2I
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
```

7. Explain which part of the attribute declaration process in the program above!

- **The declaration attribute process inside Students class like below.**

```java
public class Students {
    public int nim;
    public String name;
    public String address;
    public String class_;
```

8. Explain which part of the method declaration process in the program above!

- The method declaration process inside Students class like below.

```
Codeium: Refactor | Explain | Generate Javadoc
public void printStudents() {
    System.out.println("Nim : " + nim);
    System.out.println("Name : " + name);
    System.out.println("Address : " + address);
    System.out.println("Class : " + class_);
}
```

9. How many objects are instantiated in the program above!

- **According to my program, there is only one object that is instantiated.**

```
Students students1 = new Students();
```

10. What is actually done in the program syntax "mhs1.nim=101"?

- **The program syntax above is fill 101 at nim attribute inside mhs1 object.**

11. What is actually done in the program syntax "mhs1.tampilBiodata()"?

- **The program syntax above is a calling method that is named tampilBiodata() from mhs1 object.**

12. Instantiate 2 more objects in the program above!

Code :

```
Students students2 = new Students();
students2.nim = 456;
students2.name = "Azhar";
students2.address = "Soekarno Hatta Street";
students2.class_ = "2I";
students2.printStudents();

Students students3 = new Students();
students3.nim = 789;
students3.name = "Rizqi";
students3.address = "Soekarno Hatta Street";
students3.class_ = "2I";
students3.printStudents();
```

Result:

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
ceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-2/cod
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Nim : 123
Name : Azhar
Address : Soekarno Hatta Street
Class : 2I
Nim : 456
Name : Azhar
Address : Soekarno Hatta Street
Class : 2I
Nim : 789
Name : Rizqi
Address : Soekarno Hatta Street
Class : 2I
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
```

Experiment 3: Writing a method that has an argument/parameter and has a return

Work steps:

1. Open a text editor or IDE, such as Notepad++ / netbeans.

2. Type the following program code:

Students.java | TestStudents.java | Product.java | TestProduct.java

src > Exp3 > Product.java > Product > addStock(int)

```java
package Exp3;

public class Product {
    public String name, type;
    public int stock;

    Codeium: Refactor | Explain | Generate Javadoc
    public void printProduct() {
        System.out.println("Product name: " + name);
        System.out.println("Product type: " + type);
        System.out.println("Stock: " + stock);
    }

    // method
    Codeium: Refactor | Explain
    public int addStock(int stockIn) {
        int newStock = stockIn + stock;
        return newStock;
    }
}
```

3. Save with file name Product.java

4. To be able to access the members of an object, an instance must first be created

of the class first. The following is how to access the members of the

members of the Goods class by opening a new file then typing the following program code

program code below:

```
Students.java | TestStudents.java | Product.java | TestProduct.java ×
src > Exp3 > TestProduct.java > ...
1   package Exp3;
2
3   public class TestProduct {
        Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
4       public static void main(String[] args) {
5           Product product1 = new Product();
6           product1.name = "pencil";
7           product1.type = "ATK";
8           product1.stock = 10;
9           product1.printProduct();
10          // display and fill argument to add product stock
11          System.out.println("New stock: " + product1.addStock(stockIn:20));
12      }
13  }
14
```

5. Save with the file name TestBarang.java

6. Run the program!

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
ceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-2/code
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Product name: pencil
Product type: ATK
Stock: 10
New stock: 30
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
```

7. What is the function of arguments in a method?

- **The function of arguments is to fill as input to process a method.**

8. Draw conclusions about the use of the return keyword, and when a method must have a return!

must have a return!

- **We can use the return inside method when we create a method that has a calculation or it should have a result. But not all the methods should have returned.**

Assignment

1. A video game rental shop, one of the things that is processed is borrowing, where the data recorded when someone makes a loan is id, name, and name.

data recorded when someone borrows is id, member name, game name, and price to be paid.

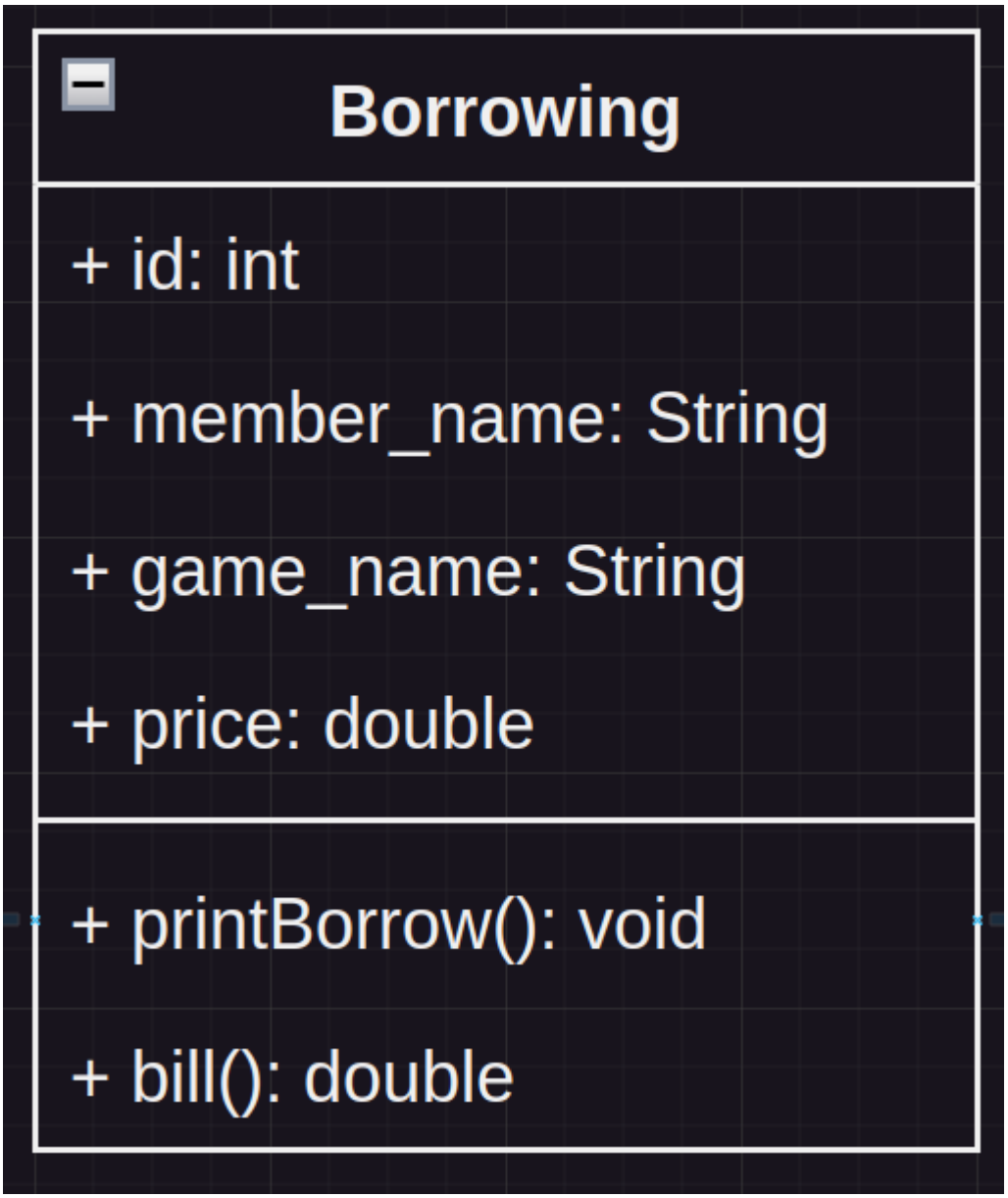
member, game name, and price to be paid. Each loan can

display data on the results of borrowing and the price to be paid. Make a class

diagram for the case study above!

Explanation:

- The price to be paid is obtained from the length of the rental x the price.
- It is assumed that 1x game loan transaction is borrowed only 1 game.



2. Make a program from the class diagram that you have made in number 1!

Code:

Borrow:

Borrow.java | BorrowMain.java

src > assnum2 > Borrow.java > Borrow > printBorrow()

```java

public class Borrow {
    public int id;
    public String member_name, game_name;
    public double price;

    Codeium: Refactor | Explain | Generate Javadoc
    public double bill(int amountDays) {
        double total = price * amountDays;
        return total;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void printBorrow() {
        System.out.println("Id : " + id);
        System.out.println("Member name : " + member_name);
        System.out.println("Game name : " + game_name);
        System.out.println("Price : " + price);
    }
}

```

Main:

Borrow.java | BorrowMain.java

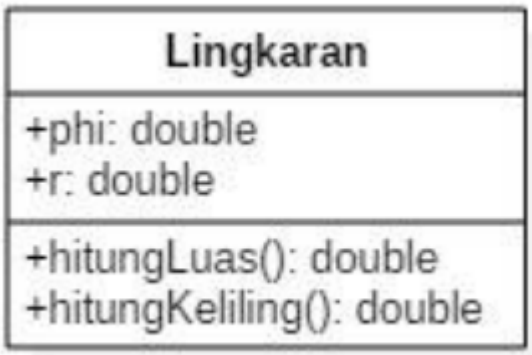
src > assnum2 > BorrowMain.java > BorrowMain > main(String[])

```java
package assnum2;

public class BorrowMain {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Borrow borrow1 = new Borrow();
        borrow1.id = 1;
        borrow1.member_name = "Azhar";
        borrow1.game_name = "GTA";
        borrow1.price = 50_000;
        borrow1.printBorrow();
        int amountDays = 1;
        System.out.println("Amount days of borrow: " + amountDays);
        System.out.println("Total bill: " + borrow1.bill(amountDays));
    }
}

```

Result:

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
ceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-2/cod
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Id : 1
Member name : Azhar
Game name : GTA
Price : 50000.0
Amount days of borrow: 1
Total bill: 50000.0
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
```

3. Make a program according to the following class diagram:



Code:

Circle.java

```java
package assnum3;

public class Circle {
    public double phi = 3.14, r;

    Codeium: Refactor | Explain | Generate Javadoc
    public double calculateArea() {
        return phi * r * r;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public double calculateCircumference() {
        return 2 * phi * r;
    }
}
```

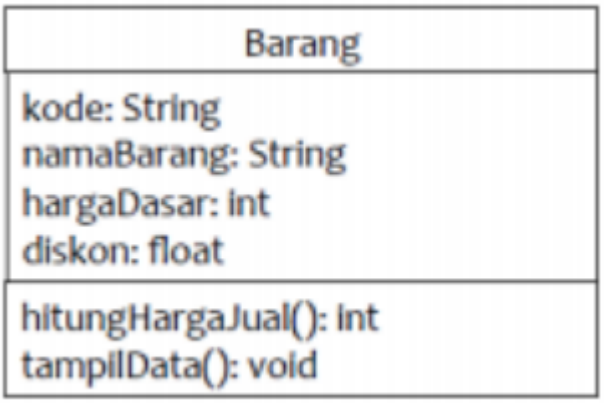
CircleMain.java

```java
package assnum3;

public class CircleMain {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Circle circle1 = new Circle();
        circle1.r = 10;
        System.out.println("Radius: " + circle1.r);
        System.out.println("Area: " + circle1.calculateArea());
        System.out.println("Circumference: " + circle1.calculateCircumference());
    }
}
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

➜ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
ceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-2/code
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Radius: 10.0
Area: 314.0
Circumference: 62.800000000000004
➜ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
```

4. Make a program according to the following class diagram:



Description / Explanation:

- Value of priceBase attribute in Rupiah and discount attribute in %
- The calculateSalePrice() method is used to calculate the sale price with the following
- the following calculation:
    - selling price = base price - (discount x base price)
- The displayData() method is used to display the values of code, itemName, base price, discount and sale price.

Code:

Product.java ProductMain.java

src > assnum4 > Product.java > ...

```java
package assnum4;

public class Product {
    public String code, name;
    public int basePrice;
    public float discount;

    Codeium: Refactor | Explain | Generate Javadoc
    public void printData() {
        System.out.println("Code: " + code);
        System.out.println("Name: " + name);
        System.out.println("Base price: " + basePrice + " IDR");
        System.out.println("Discount: " + discount + "%");
        System.out.println("Sale price: " + calculateSalePrice());
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public int calculateSalePrice() {
        float salePriceFloat = basePrice - (discount * basePrice);
        int salePrice = (int) salePriceFloat;
        return salePrice;
    }
}
```

Product.java ProductMain.java

src > assnum4 > ProductMain.java > ProductMain > main(String[])

```java
package assnum4;

public class ProductMain {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Product product1 = new Product();
        product1.code =  "123";
        product1.name = "laptop";
        product1.basePrice = 15_000_000;
        product1.discount = 0.25f;
        product1.printData();
    }
}
```

Result:

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
ceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-2/code
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Code: 123
Name: laptop
Base price: 15000000 IDR
Discount: 0.25%
Sale price: 11250000
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-2/code git:(master) ✗
```