Object Oriented Programming Inheritance

From:

AL AZHAR RIZQI RIFA'I FIRDAUS

Class:

2 I

Absence:

01

Student Number Identity:

2241720263

Department:

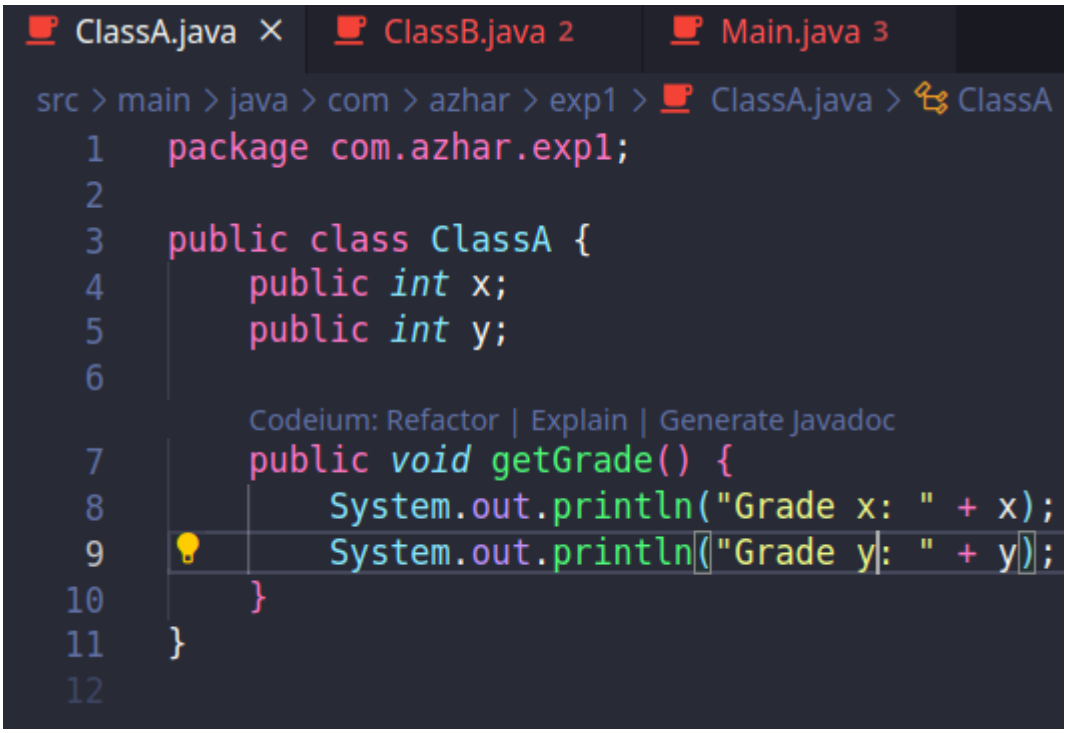
Information Technology

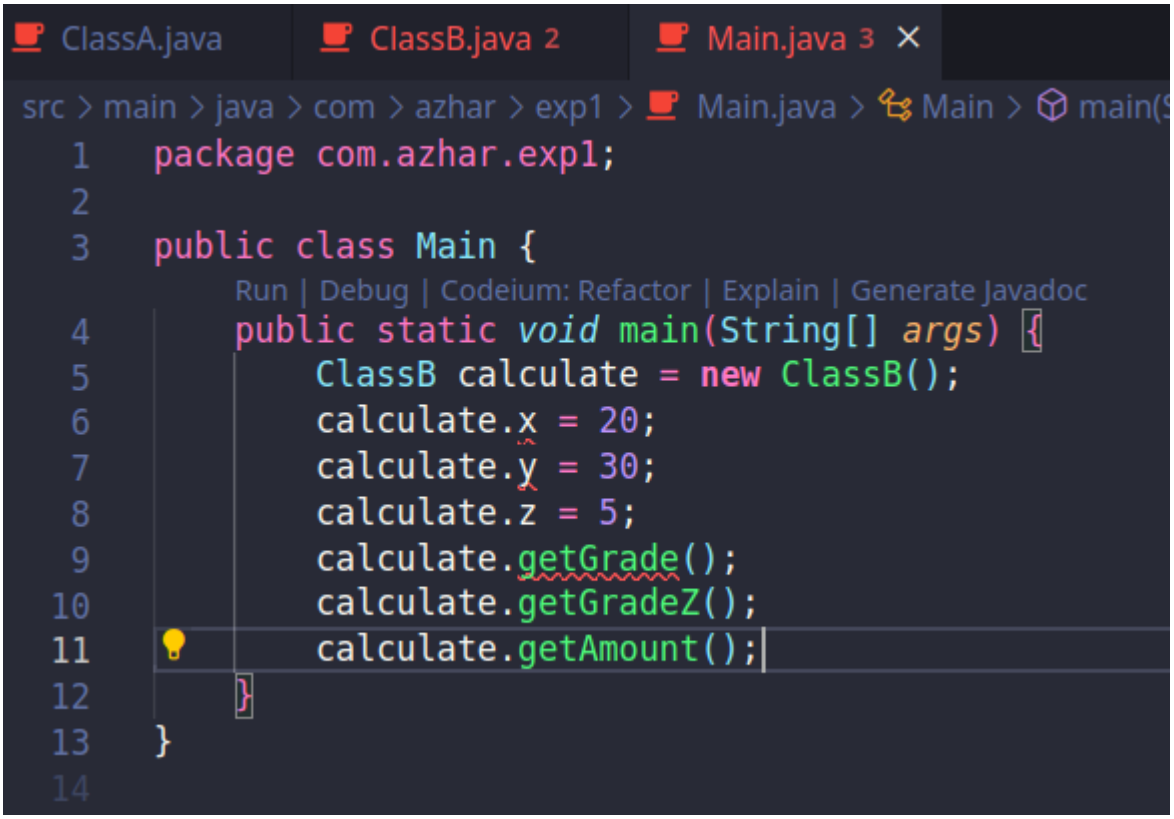
Study Program:

Informatics Engineering

Experiment 1

Coding:





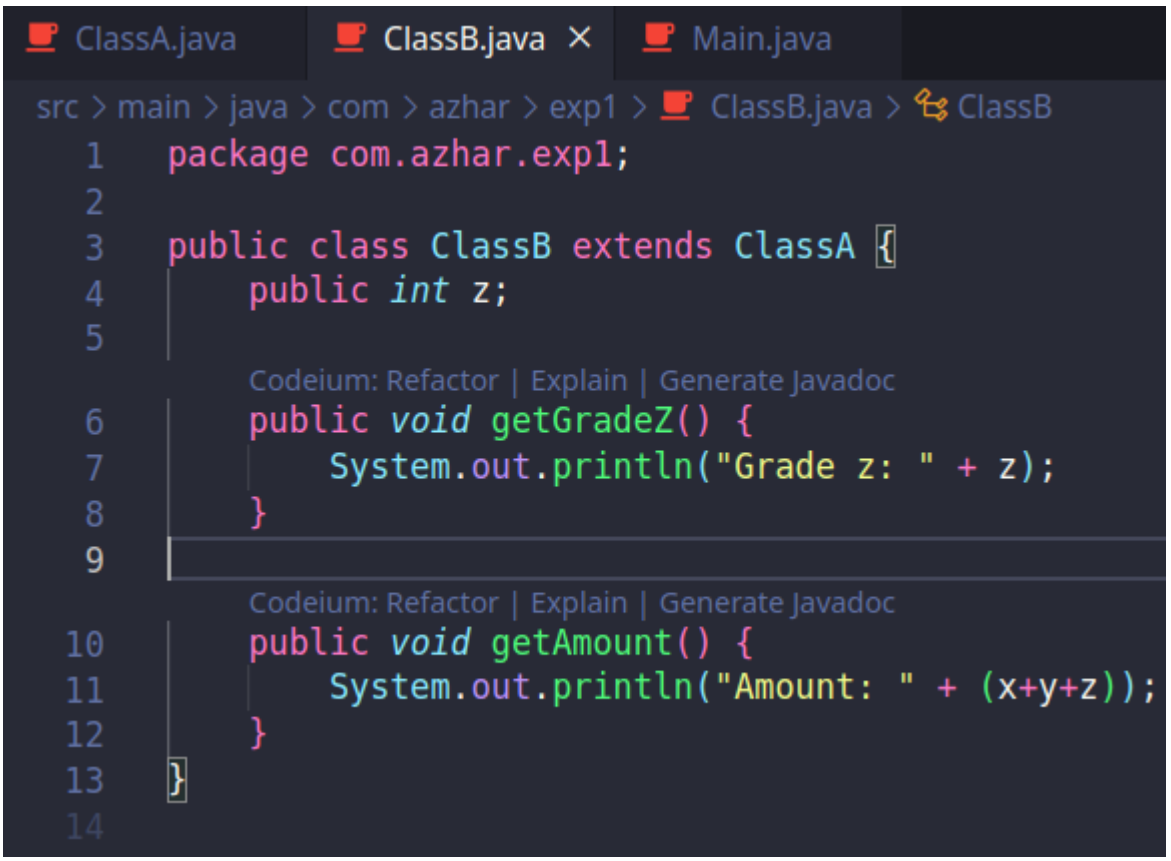
Result:

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗  /usr/bin/env /usr
 /home/zharsuke/Documents/College/Semester_3/oop/meet-5/coding/target/classes com.azhar.exp1.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
        x cannot be resolved or is not a field
        y cannot be resolved or is not a field
        The method getGrade() is undefined for the type ClassB

        at com.azhar.exp1.Main.main(Main.java:6)
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```

Question

1. In the experiment 1 above, the program that was run had an error. Then, fix it so that the program can be run and no longer has errors!



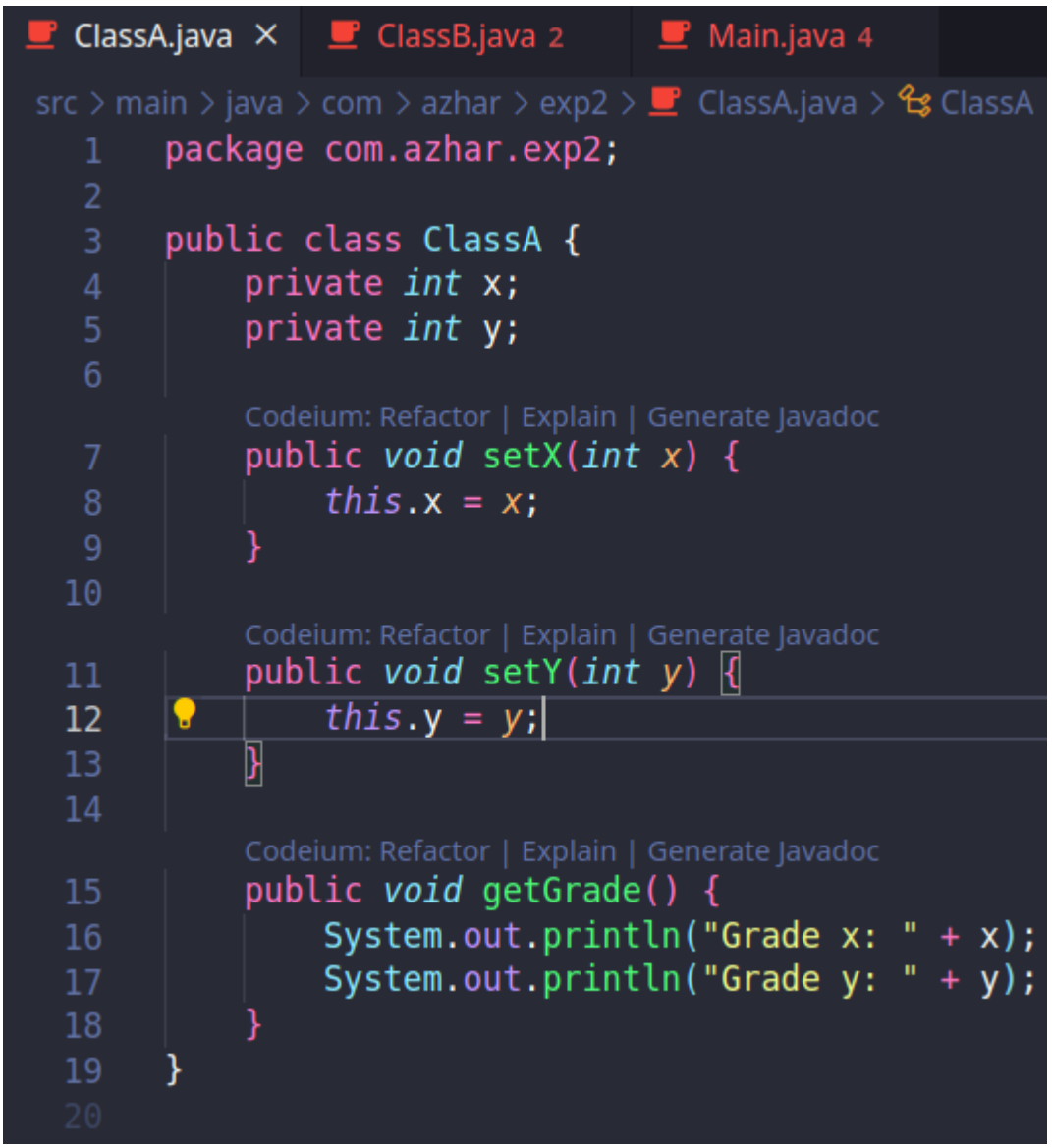

```
➜ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
xceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-5/codi
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Grade x: 20
Grade y: 30
Grade z: 5
Amount: 55
➜ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```

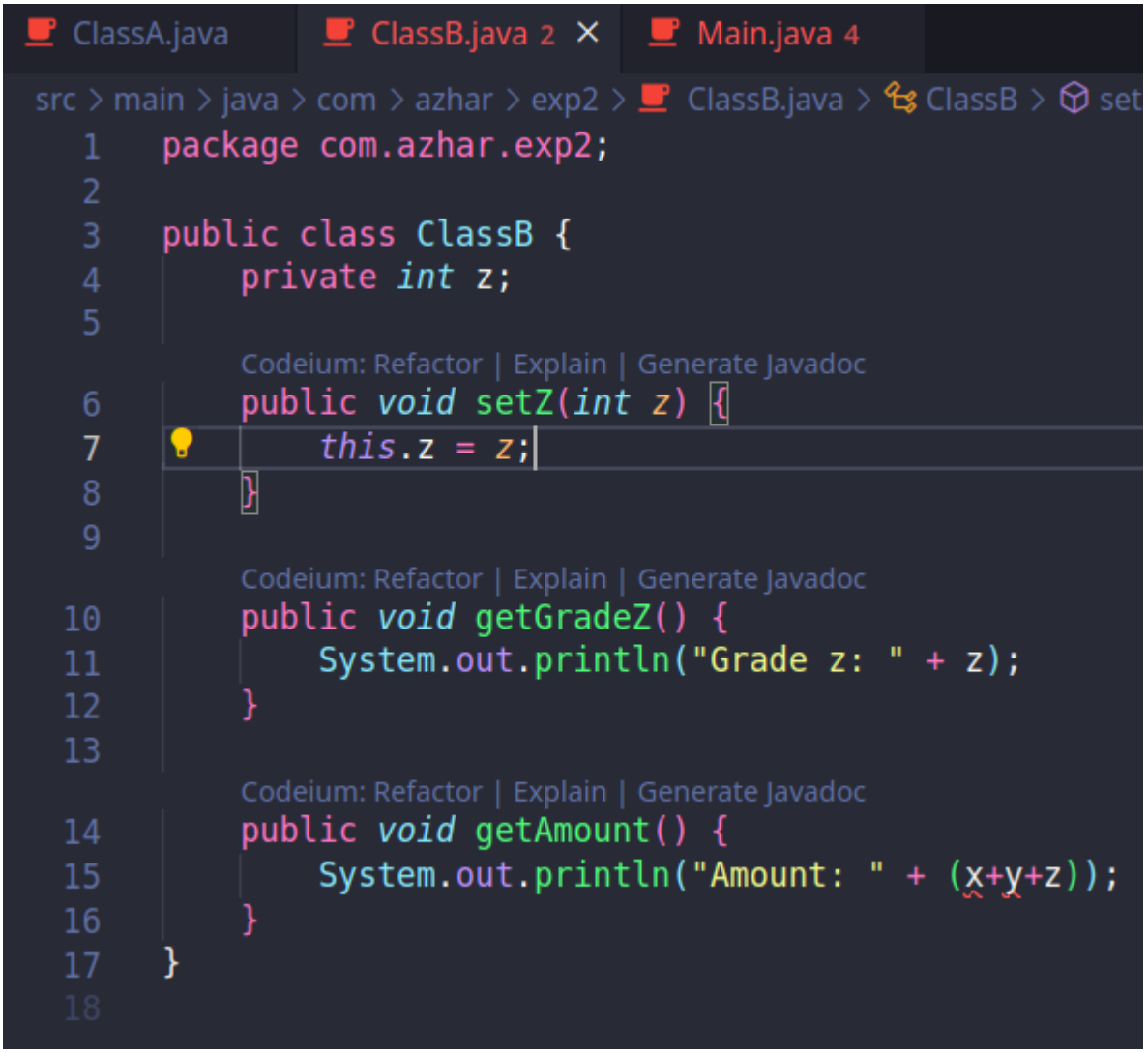
2. Explain the cause of the error when the program in experiment 1 is run!

- **From experiment above, the error was appear because we doesn't add extends in ClassB because in classB, we access ClassA attribute so that we must to extends first.**

Experiment 2

Code:


ClassA.java
ClassB.java 2
Main.java 4
src > main > java > com > azhar > exp2 > ClassA.java > ClassA
package com.azhar.exp2;
public class ClassA {
private int x;
private int y;
Codeium: Refactor | Explain | Generate Javadoc
public void setX(int x) {
this.x = x;
}
Codeium: Refactor | Explain | Generate Javadoc
public void setY(int y) {
this.y = y;
}
Codeium: Refactor | Explain | Generate Javadoc
public void getGrade() {
System.out.println("Grade x: " + x);
System.out.println("Grade y: " + y);
}
}


ClassA.java
ClassB.java 2
Main.java 4
src > main > java > com > azhar > exp2 > ClassB.java > ClassB > set
1 package com.azhar.exp2;
2
3 public class ClassB {
4 private int z;
5
Codeium: Refactor | Explain | Generate Javadoc
6 public void setZ(int z) {
7 this.z = z;
8 }
9
Codeium: Refactor | Explain | Generate Javadoc
10 public void getGradeZ() {
11 System.out.println("Grade z: " + z);
12 }
13
Codeium: Refactor | Explain | Generate Javadoc
14 public void getAmount() {
15 System.out.println("Amount: " + (x+y+z));
16 }
17 }
18

ClassA.java | ClassB.java 2 | Main.java 3

src > main > java > com > azhar > exp2 > Main.java > Main

```java
package com.azhar.exp2;

public class Main {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadc
    public static void main(String[] args) {
        ClassB calculate = new ClassB();
        calculate.setX(20);
        calculate.setY(30);
        calculate.setZ(z:5);
        calculate.getGrade();
        calculate.getGradeZ();
        calculate.getAmount();
    }
}
```

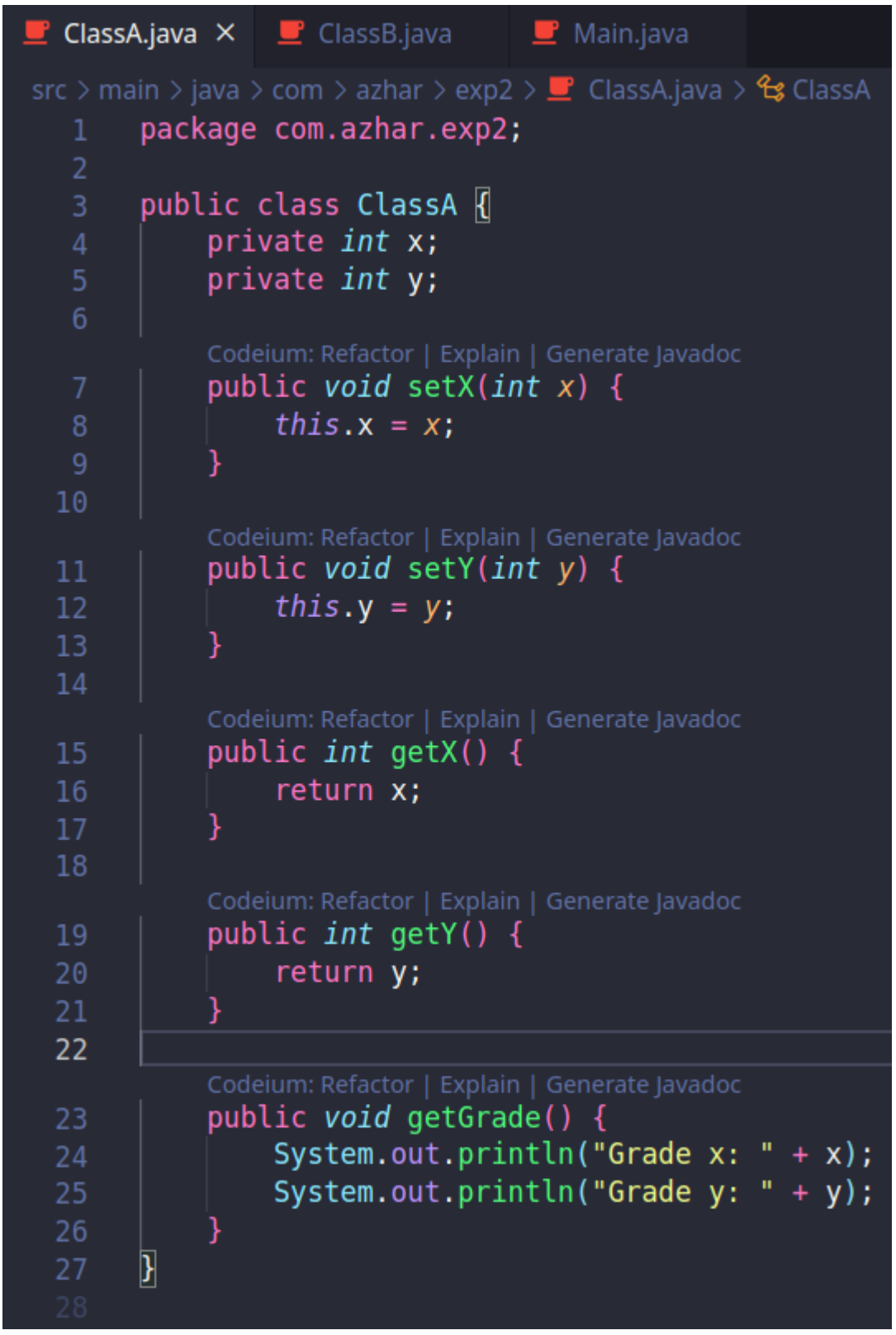
Result:

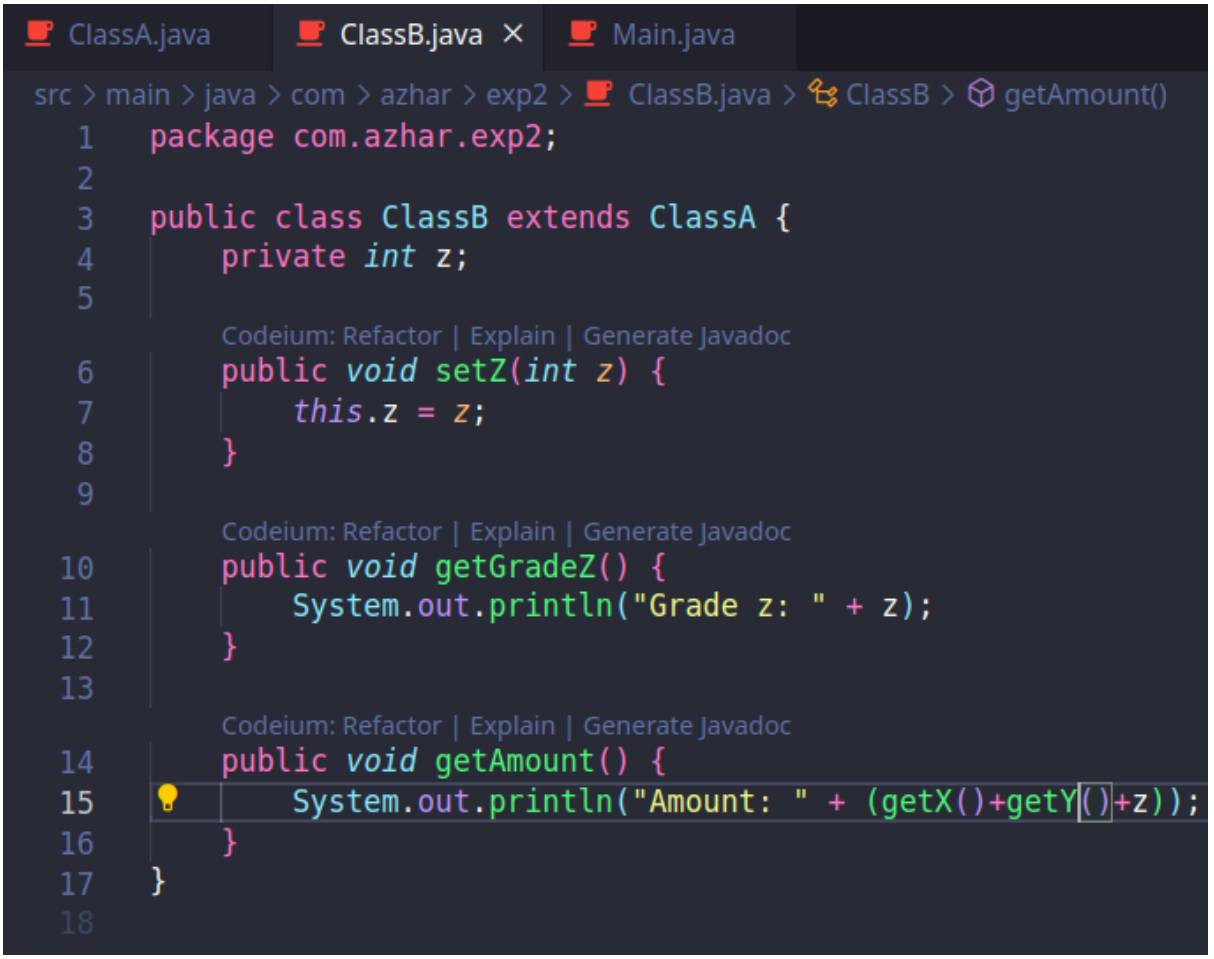
```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗  /usr/bin/env /usr
 /home/zharsuke/Documents/College/Semester_3/oop/meet-5/coding/target/classes com.azhar.exp2.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
        x cannot be resolved or is not a field
        y cannot be resolved or is not a field
        The field ClassB.z is not visible
        The method getGrade() is undefined for the type ClassB

        at com.azhar.exp2.Main.main(Main.java:6)
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```

Question

1. In experiment 2 above, an error occurred in the program being run, then fix it accordingly the program can be run and there are no errors!


ClassA.java
ClassB.java
Main.java
src > main > java > com > azhar > exp2 > ClassA.java > ClassA
package com.azhar.exp2;
public class ClassA {
private int x;
private int y;
Codeium: Refactor | Explain | Generate Javadoc
public void setX(int x) {
this.x = x;
}
Codeium: Refactor | Explain | Generate Javadoc
public void setY(int y) {
this.y = y;
}
Codeium: Refactor | Explain | Generate Javadoc
public int getX() {
return x;
}
Codeium: Refactor | Explain | Generate Javadoc
public int getY() {
return y;
}
Codeium: Refactor | Explain | Generate Javadoc
public void getGrade() {
System.out.println("Grade x: " + x);
System.out.println("Grade y: " + y);
}
}




```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
ExceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-5/cod
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Grade x: 20
Grade y: 30
Grade z: 5
Amount: 55
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```

2. Explain what caused the program in experiment 1 to have an error when it was run!

- **According code above, the error was appear because in class A we set attribute x y to private, then in clas B we try to access and it can't. The solution is we create getter method x y in class A then, call it in Class B.**

Experiment 3

Coding:

Bangun.java | Tabung.java

src > main > java > com > azhar > exp3 >

```java
package com.azhar.exp3;

public class Bangun {
    protected double phi;
    protected int r;
}
```

Bangun.java | Tabung.java | Main.java

src > main > java > com > azhar > exp3 > Tabung.java > Tabung > setT(int)

```java
package com.azhar.exp3;

public class Tabung extends Bangun {
    protected int t;

    Codeium: Refactor | Explain | Generate Javadoc
    public void setSuperPhi(double phi) {
        super.phi = phi;
    }
    Codeium: Refactor | Explain | Generate Javadoc
    public void setSuperR(int r) {
        super.r = r;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setT(int t) {
        this.t = t;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void volume() {
        System.out.println("Volume: " + (super.phi * super.r * super.r * this.t));
    }
}
```

Bangun.java | Tabung.java | Main.java

src > main > java > com > azhar > exp3 > Main.java > Main

```java
package com.azhar.exp3;

public class Main {
    Run | Debug | Codeium: Refactor | Explain | Generate Javado
    public static void main(String[] args) {
        Tabung tabung = new Tabung();
        tabung.setSuperPhi(phi:3.14);
        tabung.setSuperR(r:10);
        tabung.setT(t:3);
        tabung.volume();
    }
}
```

Result:

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗  /usr/bin/env /usr
 /home/zharsuke/Documents/College/Semester_3/oop/meet-5/coding/target/classes com.azhar.exp3.Main
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Volume: 942.0
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```

Question

1. Explain the "super" function in the following program fragment in the Tube class!

```java
public void setSuperPhi(double phi){
super.phi = phi;
}
public void setSuperR(int r){
super.r = r;
}
```

- **The function of super is with super, we can access attribute from parent class.**

2. Explain the "super" and "this" functions in the following program fragment in the Tube class!

```java
public void volume(){
System.out.println("Volume Tabung adalah: "+(super.phi*super.r*super.r*this.t));
```

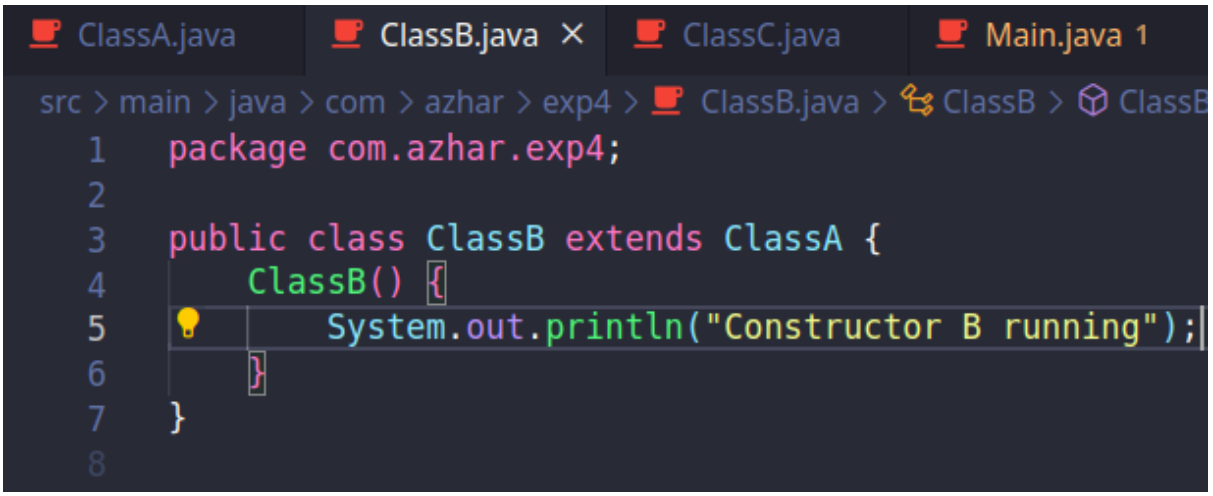
- **The super is to get attribute from parent class, then this is to get attribute local class.**

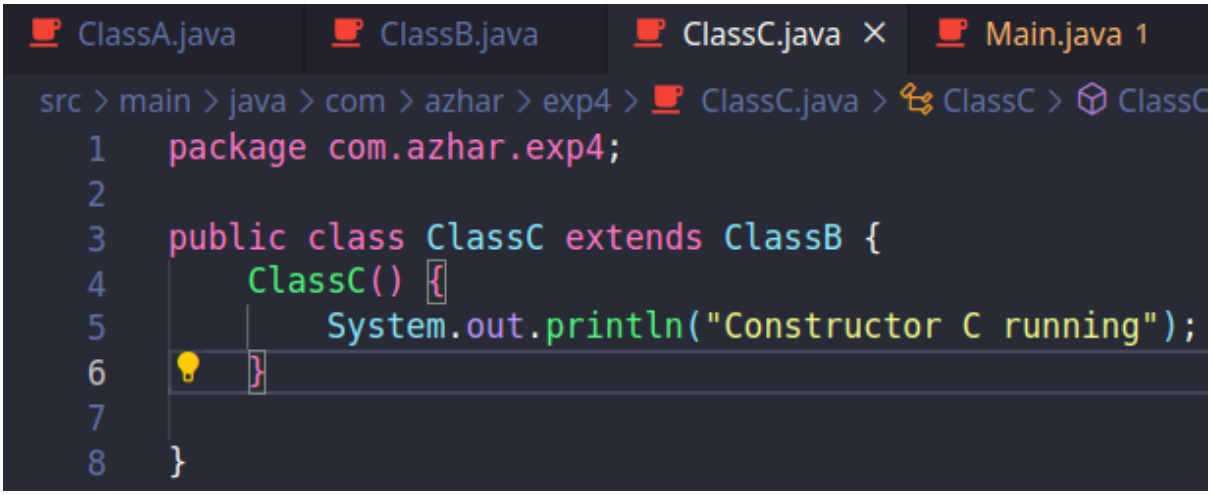
3. Explain why the Tube class does not declare the attributes "phi" and "r" but the class can access these attributes!

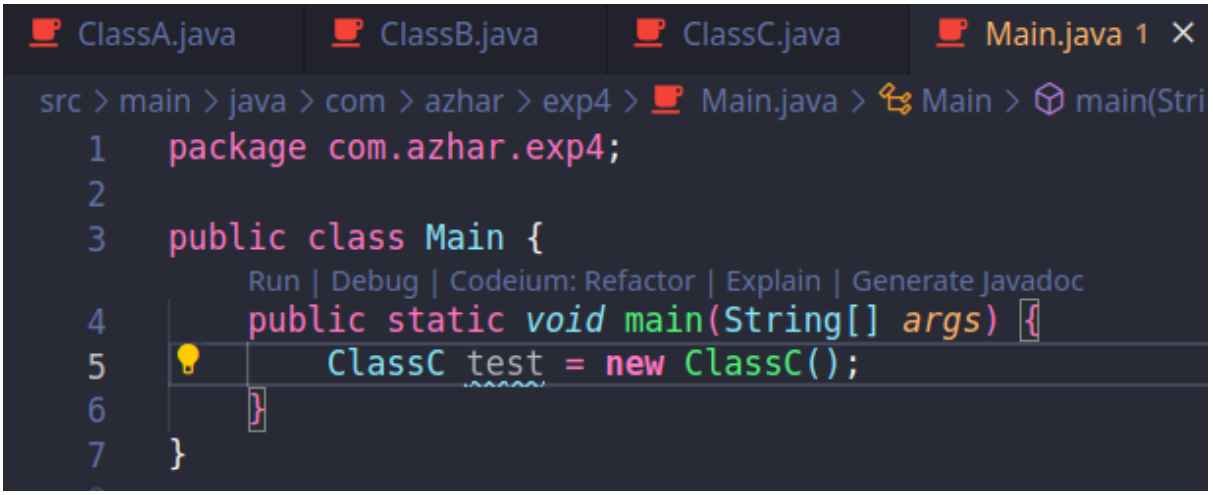
- **Because it use super to call attribute phi and r from Bangun class which is parent class of Tabung Class.**

Experiment 4

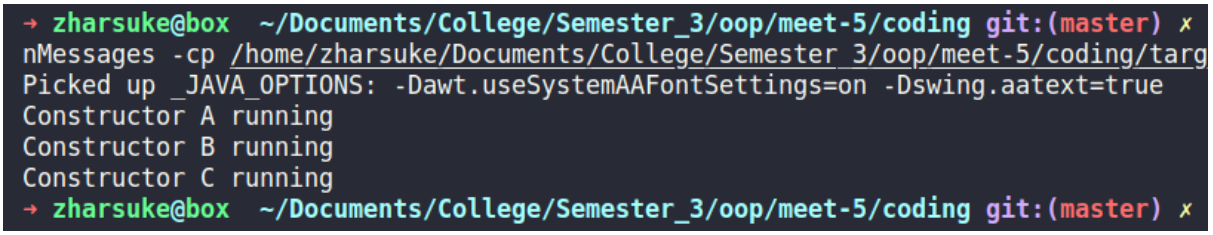
Coding:







Result:



Question

1. In experiment 4, state which classes are superclass and subclass, then explain why!

- **Based on Code above, it can be seen that class B is a subclass of class A, so in this case class A is a superclass and class B is a subclass. Then class B was the beginning is a subclass that has another subclass, namely class C, so that class B becomes the superclass of class C, as well as its rival if class C has more subclasses.**

2. Change the contents of the ClassC default constructor as follows:

```java
public class ClassC extends ClassB {
    ClassC() {
        super();
        System.out.println("Constructor C running");
    }

}
```

Add the word super() in the Garden line in the default constructor. Try running it

return the Experiment4 class and there is no difference in the output results!

3. Change the contents of the ClassC default constructor as follows:

ClassA.java | ClassB.java | ClassC.java 1

src > main > java > com > azhar > exp4 > ClassC.java > ClassC > ClassC

```java
package com.azhar.exp4;

public class ClassC extends ClassB {
    ClassC() {
        System.out.println("Constructor C running");
        super();
    }

}
```

When changing the position of super() it is on the second line in the default constructor and it appears there error. Then return super() to the first line as before, hence the error will disappear. Note the output results when the Experiment4 class is run. Why does the output appear? as follows when instantiating the test object from the ClassC class

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
ExceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-5/cod
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Constructor A running
Constructor B running
Constructor C running
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```

Explain the order in which the constructor runs when the test object is created!

- First, a ClassC test object is created using the new operator. When the test object is created, the constructor of the ClassC class will be executed first. The ClassC constructor will call the constructor of its parent class, ClassB, using the super() keyword. The ClassB constructor will call the constructor of its parent class, ClassA, using the super() keyword. The ClassA constructor will print the message "Constructor A running" to the console. The ClassB

constructor will print the message "Constructor B running" to the console. The ClassC constructor will print the message "Constructor C running" to the console.

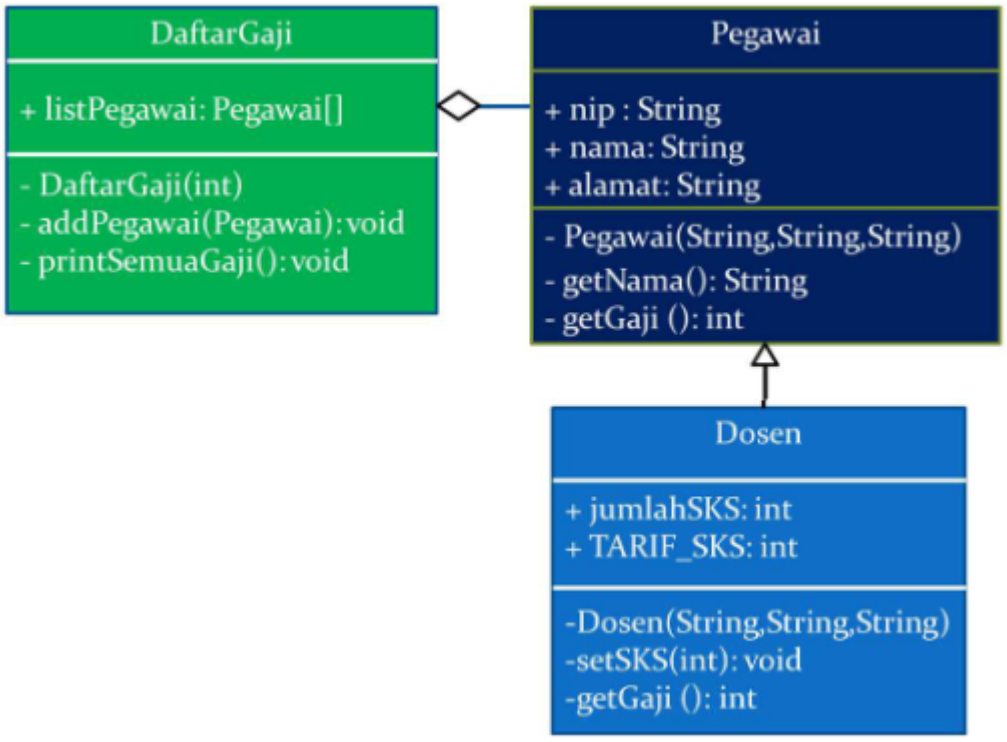
4. What is the super() function in the program snippet below in ClassC!

```
public class ClassC extends ClassB{
    ClassC(){
    super();
    System.out.println("konstruktor C dijalankan");
    }
}
```

- **It is used to call superclass methods, and to access the superclass constructor which is the constructor of ClassB.**

Assignment

1. Create a program with the concept of inheritance as in the following class diagram. Then create an object instance to display employee name and salary data got it.



Code:

Main.java | ListSalary.java | Employee.java | Lecture.java

src > main > java > com > azhar > asg > ListSalary.java > ListSalary > employee

```java
package com.azhar.asg;

public class ListSalary {
    private Employee employee[];

    public ListSalary(int size) {
        this.employee = new Employee[size];
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void addEmployee(Employee employee) {
        for (int i = 0; i < this.employee.length; i++) {
            if (this.employee[i] == null) {
                this.employee[i] = employee;
                break;
            }
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void printAllSalary() {
        System.out.println("==========List Salary of All Employees==========");
        for (int i = 0; i < employee.length; i++) {
            System.out.println("Salary of " + employee[i].getName() + " = " + employee[i].getSalary());
        }
        System.out.println("================================================");
    }
}
```

Main.java | ListSalary.java | Employee.java | Lecture.java

src > main > java > com > azhar > asg > Employee.java > Employee > setName(String

```java
package com.azhar.asg;

public class Employee {
    private String nip;
    private String name;
    private String address;

    public Employee() {

    }

    public Employee(String nip, String name, String address) {
        this.nip = nip;
        this.name = name;
        this.address = address;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setNip(String nip) {
        this.nip = nip;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setAddress(String address) {
        this.address = address;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setName(String name) {
        this.name = name;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getNip() {
        return nip;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getAddress() {
        return address;
    }

```

```
        Codeium: Refactor | Explain | Generate Javadoc
38      public String getName() {
39          return name;
40      }
41
        Codeium: Refactor | Explain | Generate Javadoc
42      public int getSalary() {
43          return 0;
44      }
45  }
46
```


Main.java
ListSalary.java
Employee.java
Lecture.java
src > main > java > com > azhar > asg > Lecture.java > Lecture > setRatesSKS
1 package com.azhar.asg;
2
3 public class Lecture extends Employee {
4 private int amountSKS;
5 private int ratesSKS;
6
7 public Lecture() {
8 super();
9 }
10
Codeium: Refactor | Explain | Generate Javadoc
11 public void setAmountSKS(int amountSKS) {
12 this.amountSKS = amountSKS;
13 }
14
Codeium: Refactor | Explain | Generate Javadoc
15 public void setRatesSKS(int ratesSKS) {
16 this.ratesSKS = ratesSKS;
17 }
18
Codeium: Refactor | Explain | Generate Javadoc
19 public int getSalary() {
20 return amountSKS * ratesSKS;
21 }
22
23 }
24

Main.java | ListSalary.java | Employee.java | Lecture.java

src > main > java > com > azhar > asg > Main.java > Main > main(String[])

```java
package com.azhar.asg;

public class Main {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        ListSalary listSalary = new ListSalary(size:2);
        Lecture lecture1 = new Lecture();
        lecture1.setNip(nip:"1234");
        lecture1.setName(name:"Azhar");
        lecture1.setAddress(address:"Jl. Sukarno Hatta No.9");
        lecture1.setAmountSKS(amountSKS:5);
        lecture1.setRatesSKS(ratesSKS:100_000);

        Lecture lecture2 = new Lecture();
        lecture2.setNip(nip:"5678");
        lecture2.setName(name:"Rizqi");
        lecture2.setAddress(address:"Jl. Sukarno Hatta No.9");
        lecture2.setAmountSKS(amountSKS:10);
        lecture2.setRatesSKS(ratesSKS:150_000);

        listSalary.addEmployee(lecture1);
        listSalary.addEmployee(lecture2);

        listSalary.printAllSalary();

    }
}
```

Result:

```
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
ExceptionMessages -cp /home/zharsuke/Documents/College/Semester_3/oop/meet-5/cod
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
==========List Salary of All Employees==========
Salary of Azhar = 500000
Salary of Rizqi = 1500000
================================================
→ zharsuke@box  ~/Documents/College/Semester_3/oop/meet-5/coding git:(master) ✗
```