Group A

1. Which feature of OOP indicates code reusability?
   Abstraction
   Polymorphism
   Encapsulation
√ Inheritance

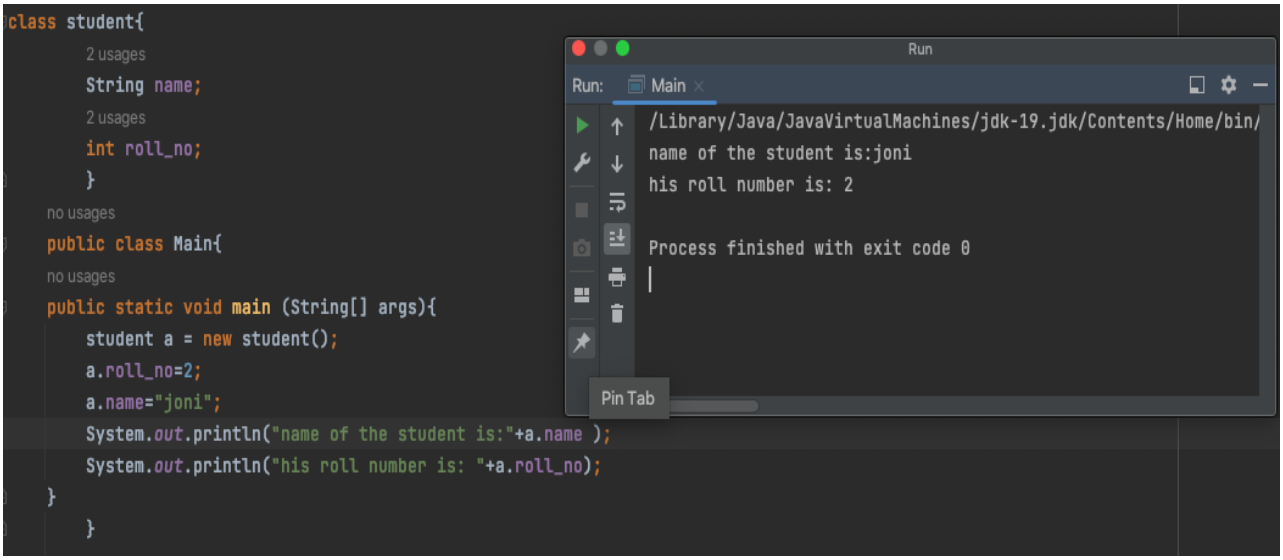
2. Which of the following is not an access modifier?
√ Abstract
   Public
   Private
   Protected

3. Encapsulation is a way of combining both data members and member functions, which operate on those data members, into a single unit. We call it a class in OOP generally. This feature have helped us modify the structures used in C language to be upgraded into class in JAVA and other languages. Show the process encapsulation in JAVA.

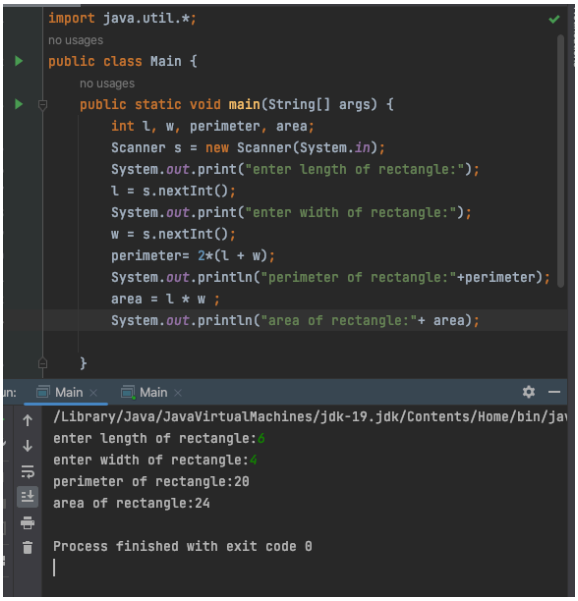
```java
public class person{
    private String name;
    private int age;
    //getter and setter methods for name
    public String getName(){
        return this.name;
    }
    public void setNmae(String name){
        this.name=name;
    }
    //getter and setter methods for age
    public int getAge(){
        return this.age;
    }
    public void setAge(int age){
        this.age=age;
    }
}
```

Group B

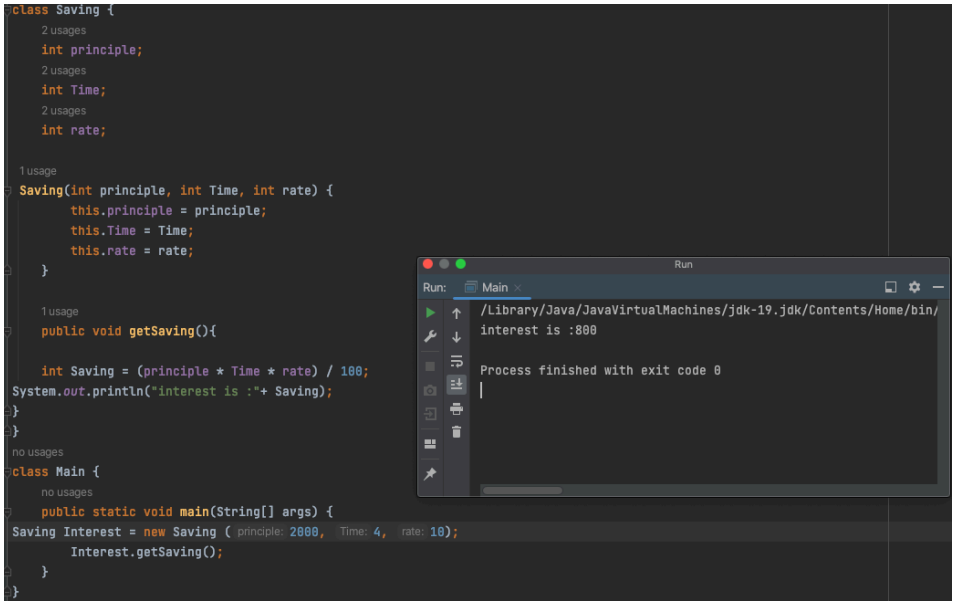
1. Create a class named 'Student' with String variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as '2' and that of name as "John" by creating an object of the class Student.



2. A rectangle has the length of 6 centimeters and width 4cm. Create a method each to print the area and perimeter of the given rectangle.



3. Write a program to print the simple interest by creating a class named “Savings” taking the values of its Principle, Time and Rate as parameters of a method named “Interest”.

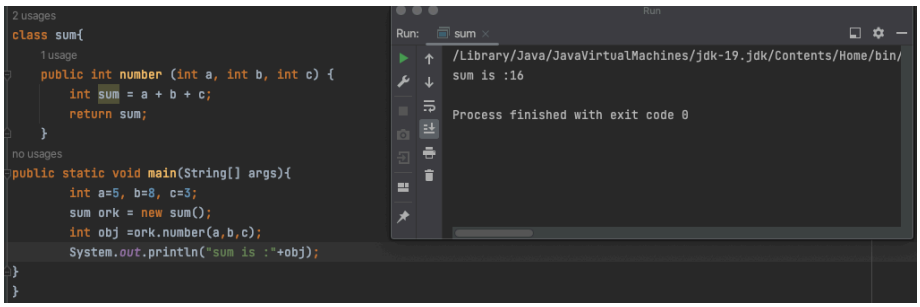


```
class Saving {
    2 usages
    int principle;
    2 usages
    int Time;
    2 usages
    int rate;

    1 usage
    Saving(int principle, int Time, int rate) {
        this.principle = principle;
        this.Time = Time;
        this.rate = rate;
    }

    1 usage
    public void getSaving(){

    int Saving = (principle * Time * rate) / 100;
System.out.println("interest is :"+ Saving);
}
}
no usages
class Main {
    no usages
    public static void main(String[] args) {
Saving Interest = new Saving ( principle: 2000,  Time: 4,  rate: 10);
        Interest.getSaving();
    }
}
```

```
Run
Run:  Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/
interest is :800

Process finished with exit code 0
```

Group C

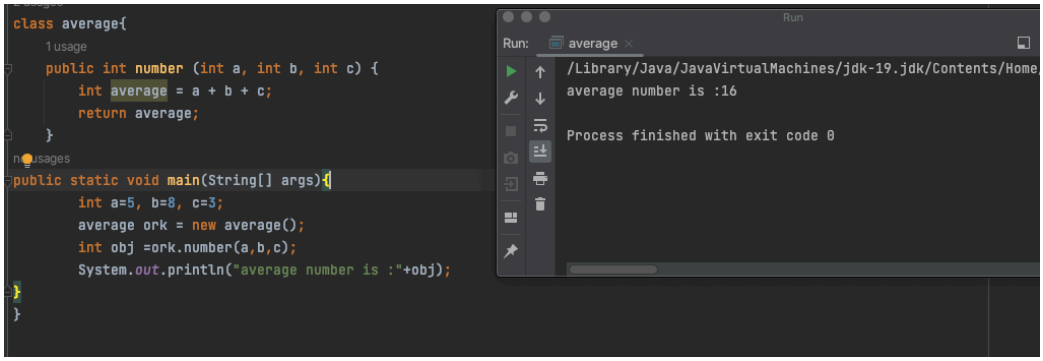
1. Write a program to find the sum of three numbers. Create a method findSum() of integer return type to print the sum.



```
2 usages
class sum{
    1 usage
    public int number (int a, int b, int c) {
        int sum = a + b + c;
        return sum;
    }
no usages
public static void main(String[] args){
        int a=5, b=8, c=3;
        sum ork = new sum();
        int obj =ork.number(a,b,c);
        System.out.println("sum is :"+obj);
}
}
```

```
Run:  sum
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/
sum is :16

Process finished with exit code 0
```

2. Write the program to find the average of three input numbers by using a method returning a double value.



```
class average{
    1 usage
    public int number (int a, int b, int c) {
        int average = a + b + c;
        return average;
    }
no usages
public static void main(String[] args){
        int a=5, b=8, c=3;
        average ork = new average();
        int obj =ork.number(a,b,c);
        System.out.println("average number is :"+obj);
}
}
```

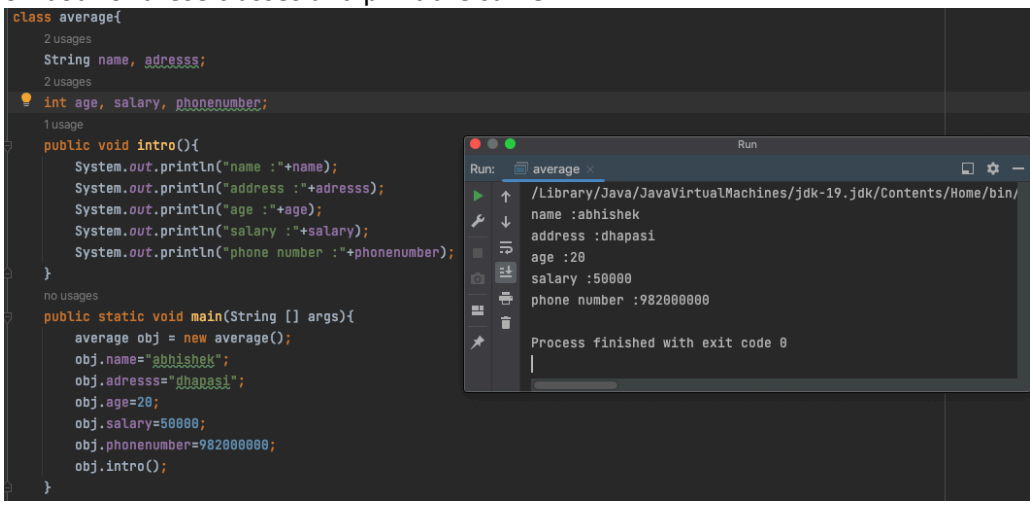
```
Run:  average
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home
average number is :16

Process finished with exit code 0
```

3. Create a class named 'Employee' having the following members:

Data members:

- Name
- Age
- Phone number
- Address
- Salary

It also has a method named 'printSalary' which prints the salary of the members. Now, assign name, age, phone number, address and salary to an employee by making an object of both of these classes and print the same.



```
class average{
    2 usages
    String name, adressss;
    2 usages
    int age, salary, phonenumber;
    1 usage
    public void intro(){
        System.out.println("name :"+name);
        System.out.println("address :"+adressss);
        System.out.println("age :"+age);
        System.out.println("salary :"+salary);
        System.out.println("phone number :"+phonenumber);
    }
    no usages
    public static void main(String [] args){
        average obj = new average();
        obj.name="abhishek";
        obj.adressss="dhapasi";
        obj.age=20;
        obj.salary=50000;
        obj.phonenumber=982000000;
        obj.intro();
    }
```

```
Run
Run: average
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/
name :abhishek
address :dhapasi
age :20
salary :50000
phone number :982000000

Process finished with exit code 0
```

Group D

1. Create a simple calculator program using java OOP.
    a. Take two non-zero inputs.
    b. Create a method to print sum, difference, product and quotient.
    c. Ask the user to choose between options (1-4) for sum, difference, product and divide operations.
    d. Give the user choice of another operation.