Group A

1. Which feature of OOP indicates code reusability?
   a) Abstraction
   b) Polymorphism
   c) Encapsulation
   d) **Inheritance**

2. Which of the following is not an access modifier?
   a) **Abstract**
   b) Public
   c) Private
   d) 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 EncapsulationExample {
    private unt age;
    private String name;
    //Getter and setter methods for age
    public int getAge() {
        return age;
        //Nabodip Thapa
    }
    public void setAge(int age) {
        this.age = age;
    }
    // Getter and setter methods for name
    public String getName() {
        return name;
    }
    public void setName(string name) {
        this.name =name;
    }
}
```

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.

```java
//Nabodip Thapa
class student {
    int rollno=21;
    String name="john";
    public static void main (String[] args){
        student stud1=new student();
        System.out.println(stud1.name);
        System.out.println(stud1.rollno);
    }

}
```

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.

w.java > rect

```java
//Nabodip Thapa
class info {
    int l =5, b =4;
    public void calculation(){
        int area = l*b ;
        int perimeter = 2* (l+b);
        System.out.println("Area is" +area);
        System.out.println("Perimeter is"+ perimeter);
    }
}
//Nabodip Thapa
class rect{
    Run | Debug
    public static void main(String[] args) {
        info obj = new info ();
        obj.calculation();
    }
}
```

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”.

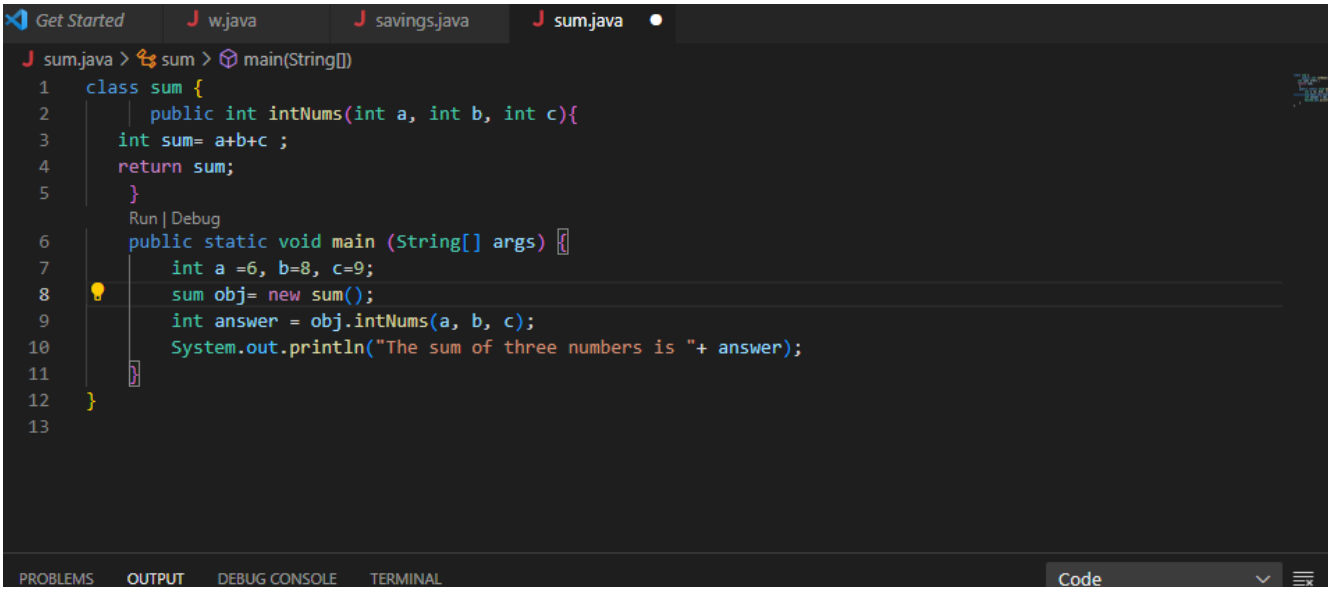
savings.java > Savings

```java
//Nabodip Thapa
class Savings{
    int Principle, Time, Rate;
    public void Interest(){
        int Si = (Principle * Time * Rate)/100;
    System.out.println("Simple Interest is "+Si);
    }
    Run | Debug
    public static void main ( String[]args){
Savings obj = new Savings();
obj.Principle = 2000;
obj.Time = 4;
obj. Rate = 50;
obj.Interest();
    }

}
```

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

```java
class avg {
    public int putNumbers(int a, int b, int c){
        int average = (a + b + c)/3;
        return average;
    }
    Run | Debug
    public static void main(String[] args) {
        int a=2, b=9, c=9;
        avg obj = new avg();
        double answer = obj.putNumbers(a, b, c);
        System.out.println("The average of three numbers is"+ answer);
    }
    }
```

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.

```java
//Nabodip Thapa
class Employee {
    String Name,Address;
    int age, Salary;
    int P_num;
    public void printSalary(){
        System.out.println("Name:"+Name);
        System.out.println("Address:"+Address);
        System.out.println("Age:"+age);
        System.out.println("Phone Number:"+P_num);
        System.out.println("Salary:"+Salary);
    }
    Run | Debug
    public static void main(String[]args) {
        Employee obj = new Employee();
        obj.Name ="Roshan";
        obj.Address="Newroad";
        obj.age= 25;
```

```
18          obj.P_num = 980531783;
19          obj.Salary = 2;
20          obj.printSalary();
21
22      }
23  }
24
```

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.

Calculator.java > Calculator > main(String[])

```
1   import java.util.*;
2   public class Calculator{
        Run | Debug
3       public static void main(String[] args){
4           Scanner sc= new Scanner(System.in);
5           System.out.println("Enter two non-zero inputs.");
6           int num1= sc.nextInt();
7           int num2= sc.nextInt();
8           while(num1>=0 && num2>=0){
9           System.out.println("Choose  number for operation:1(add), 2(subtract), 3(multiply), 4(quotient)");
10          int num3= sc.nextInt();
11          if(num3==1){
12              System.out.println("Sum of 2 numberis: " + findsum(num1, num2)); }
13              else if(num3==2){
14               System.out.println("Difference of 2 number is: " + finddiff(num1, num2));
15              }
16              else if(num3==3){
17               System.out.println("Product of 2 number is: " + findpro(num1, num2));
```

Calculator.java > Calculator > main(String[])

```
16                  else if(num3==3){
17                   System.out.println("Product of 2 number is: " + findpro(num1, num2));
18                  }
19                  else if(num3==4){
20                   System.out.printf("Quotient of 2 number is: " +"%.2f\n", findquo(num1, num2));
21                  }
22                  else{
23                      System.out.println("invalid input");
24                  }
25              System.out.println("Enter another two non-zero inputs.");
26               num1= sc.nextInt();
27               num2= sc.nextInt();
28              }
29                  System.out.println("Calculator has been stopped due to immesarable value.");
30          }
31          public static int findsum(int num1, int num2){
32              int sum= num1+num2;
33              return sum;
```

```
35          public static int finddiff(int num1, int num2){
36              int diff= num1-num2;
37              return diff;
38          }
39          public static int findpro(int num1, int num2){
40              int pro= num1*num2;
41              return pro;
42          }
43          public static double findquo(int num1, int num2){
44              double quo= (double) num1/num2;
45              return quo;
46          }
47
48      }
```