

**Iqra National University**

**Object Oriented Programming (Lab)**

**Name: Nooriya Ilyas Khan**

**ID: 15770**

**BS(SE) SEMESTER 2<sup>nd</sup>**

**Submitted To: Sir Ayub**

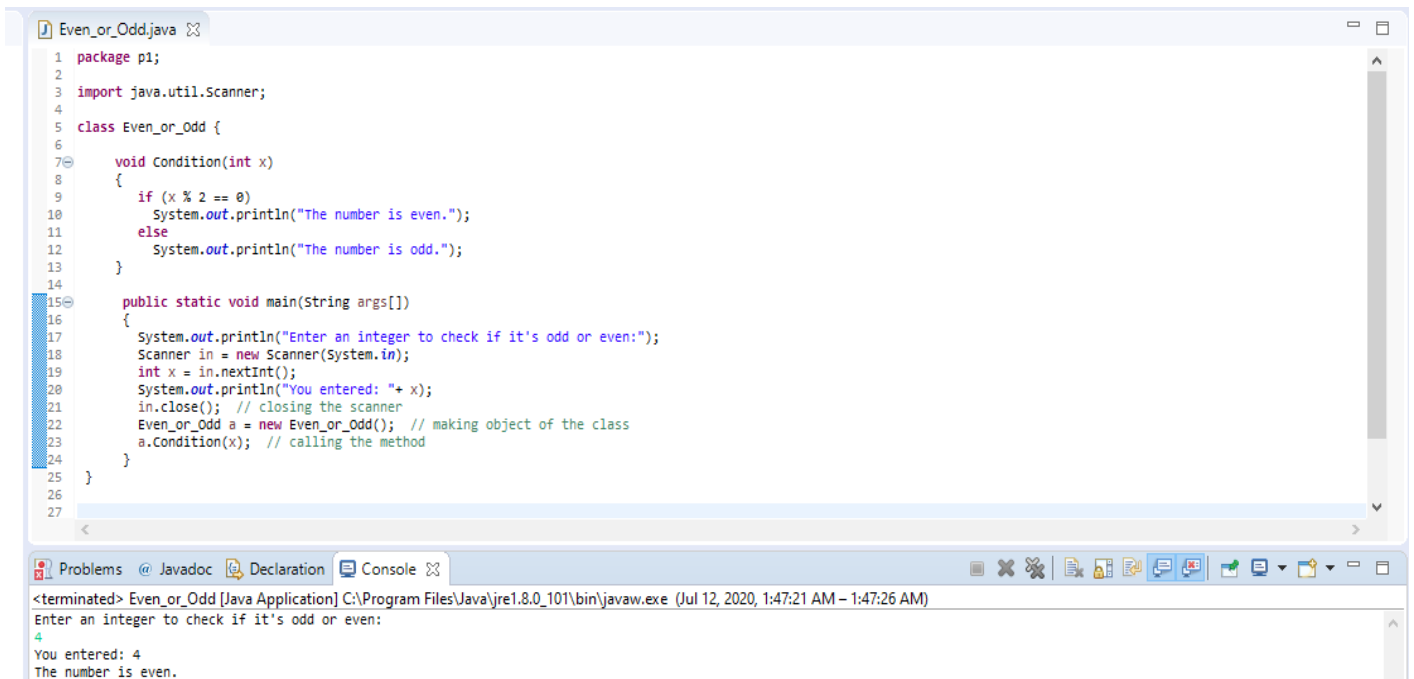
## Question1. How to check Even and Odd numbers in java using object oriented approach?

### Answer:

The program below consists of the code required to determine whether a number entered by the user is even or odd.

### Explanation:

- The class Even\_or\_Odd consists of a void method named “Condition” which takes the parameter integer x, which would be eventually entered by the user. This method has the if-else statement which simply puts forth the condition that the remainder of the user’s input (x) after dividing it with 2 would have to result in zero in order for it to be an even number, or else it would be an odd number. The modulo operator I used in the 9<sup>th</sup> line does the checking of the remainder.
- The 2<sup>nd</sup> method I have here in the class is the main method. The main method contains the statement that asks the user to enter an integer of the user’s choice, and the scanner class, which is used to take the input from user. The nextInt() method is a built-in method of the scanner class which is used to scan the next token of input as an integer. Therefore if we enter an alphabet or any other character as the input instead of an integer, we will get an error. In the last two steps I simply made an object of the class Even\_or\_Odd and called the void method that had the if-else statement in it. The result is displayed in the console.



```
1 package p1;
2
3 import java.util.Scanner;
4
5 class Even_or_Odd {
6
7     void Condition(int x)
8     {
9         if (x % 2 == 0)
10            System.out.println("The number is even.");
11        else
12            System.out.println("The number is odd.");
13    }
14
15    public static void main(String args[])
16    {
17        System.out.println("Enter an integer to check if it's odd or even:");
18        Scanner in = new Scanner(System.in);
19        int x = in.nextInt();
20        System.out.println("You entered: "+ x);
21        in.close(); // closing the scanner
22        Even_or_Odd a = new Even_or_Odd(); // making object of the class
23        a.Condition(x); // calling the method
24    }
25 }
26
27
```

Problems @ Javadoc Declaration Console

<terminated> Even\_or\_Odd [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jul 12, 2020, 1:47:21 AM – 1:47:26 AM)

Enter an integer to check if it's odd or even:

4

You entered: 4

The number is even.

**Question2.** How to add 2 complex numbers in java using object oriented approach?

**Answer:**

**Explanation:**

- The following program consists of 2 classes, first one named Main, and the other one ComplexNumber. The ComplexNumber class contains methods that would perform the required operation we want to perform on the two complex numbers, I will explain it later.
- In the main method, I created two instances (objects) of complex number 1 and 2 respectively and passed arguments into it, the first numbers 2 and 3 are the real part of the complex number while 4 and 5 would be the imaginary part of the complex number. The 'sum' stores the result of the addition of the complex numbers c1 and c2.
- The ComplexNumber class has two instance variables 'real' and 'imaginary' to hold the real and imaginary parts of the complex number respectively.
- The keyword this and the dot operator here 'this.real' and 'this.imaginary' refers to the instance variables. The method ComplexNumber sum() returns the result of the addition of the real parts of the complex numbers and the addition of imaginary parts. The word 'other' I used here is not a keyword, it is just a parameter name and in this case I used it for comparison.
- The method getReal() returns the real part of the complex number. The method getImaginary() returns imaginary part of the complex number.
- The method toString() gives a string which returns both the real and imaginary parts of the complex number, and overrides toString present in class java.lang.Object.
- Now in the main method, the code will be executed and the console will display the two complex numbers I assigned, and their sum value.

**Program with output:**

```
Main.java
1 package p1;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         ComplexNumber c1 = new ComplexNumber(2, 4); // first complex number
8         ComplexNumber c2 = new ComplexNumber(3, 5); // second complex number
9
10        ComplexNumber sum = c1.sum(c2);
11
12        System.out.println("first complex number: " + c1);
13        System.out.println("second complex number: " + c2);
14        System.out.println("sum of two complex numbers: " + sum);
15
16    }
17 }
18
19 /*
20  * This class represents a complex number. A complex number has two parts, real
21  * and imaginary.
22  */
23 class ComplexNumber {
24     double real;
25     double imaginary;
26
27     public ComplexNumber(double real, double imaginary) {
28         this.real = real;
29         this.imaginary = imaginary;
30     }
31
32     public ComplexNumber sum(ComplexNumber other) {
33         double r = this.real + other.real;
34         double i = this.imaginary + other.imaginary;
35         return new ComplexNumber(r, i);
36     }
37
38     public double getReal() {
39         return real;
40     }
41 }
```

```
15 }
16 }
17 }
18 /*
19  * This class represents a complex number. A complex number has two parts, real
20  * and imaginary.
21  */
22 class ComplexNumber {
23     double real;
24     double imaginary;
25
26     public ComplexNumber(double real, double imaginary) {
27         this.real = real;
28         this.imaginary = imaginary;
29     }
30
31     public ComplexNumber sum(ComplexNumber other) {
32         double r = this.real + other.real;
33         double i = this.imaginary + other.imaginary;
34         return new ComplexNumber(r, i);
35     }
36
37     public double getReal() {
38         return real;
39     }
40
41     public double getImaginary() {
42         return imaginary;
43     }
44
45     @Override
46     public String toString() {
47         return real + " + " + imaginary + "i";
48     }
49 }
```

Problems @ Javadoc Declaration Console

```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 12, 2020, 1:56:04 AM - 1:56:07 AM)
first complex number: 2.0 + 4.0i
second complex number: 3.0 + 5.0i
sum of two complex numbers: 5.0 + 9.0i
```

### Question3. How to check Leap year in java using object oriented approach?

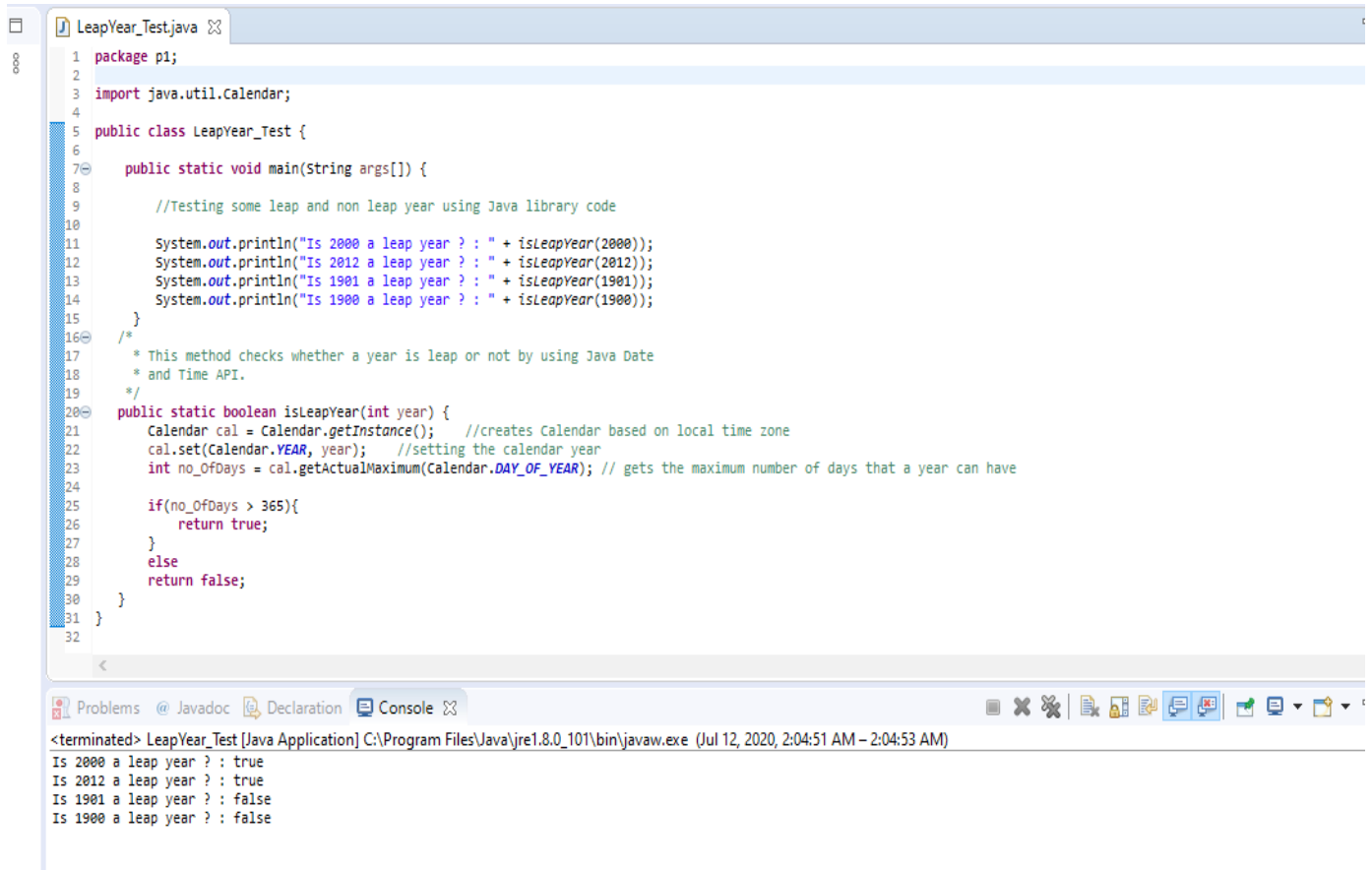
#### Answer:

I had some confusion over the two ways I could write this code, so I will be explaining the both ways I checked if a year is a leap year or not.

#### Explanation 1:

- In this code, I imported the java calendar class, an abstract class that provides methods for setting of calendar fields such as month, year etc. As it is an Abstract class, so we cannot use a constructor to create an instance. Instead, we will have to use the static method `Calendar.getInstance()` to instantiate and implement a subclass.
- Calendar class has utility method `getMaximum()` to return maximum number of days in a year which can be used to check if it's greater than 365 or not. If it has more than 365 days, it means it's a leap year.

#### Program 1 with output:



```
LeapYear_Test.java
1 package p1;
2
3 import java.util.Calendar;
4
5 public class LeapYear_Test {
6
7     public static void main(String args[] ) {
8
9         //Testing some leap and non leap year using Java library code
10
11         System.out.println("Is 2000 a leap year ? : " + isLeapYear(2000));
12         System.out.println("Is 2012 a leap year ? : " + isLeapYear(2012));
13         System.out.println("Is 1901 a leap year ? : " + isLeapYear(1901));
14         System.out.println("Is 1900 a leap year ? : " + isLeapYear(1900));
15     }
16     /*
17      * This method checks whether a year is leap or not by using Java Date
18      * and Time API.
19      */
20     public static boolean isLeapYear(int year) {
21         Calendar cal = Calendar.getInstance(); //creates Calendar based on local time zone
22         cal.set(Calendar.YEAR, year); //setting the calendar year
23         int no_ofDays = cal.getActualMaximum(Calendar.DAY_OF_YEAR); // gets the maximum number of days that a year can have
24
25         if(no_ofDays > 365){
26             return true;
27         }
28         else
29             return false;
30     }
31 }
32
```

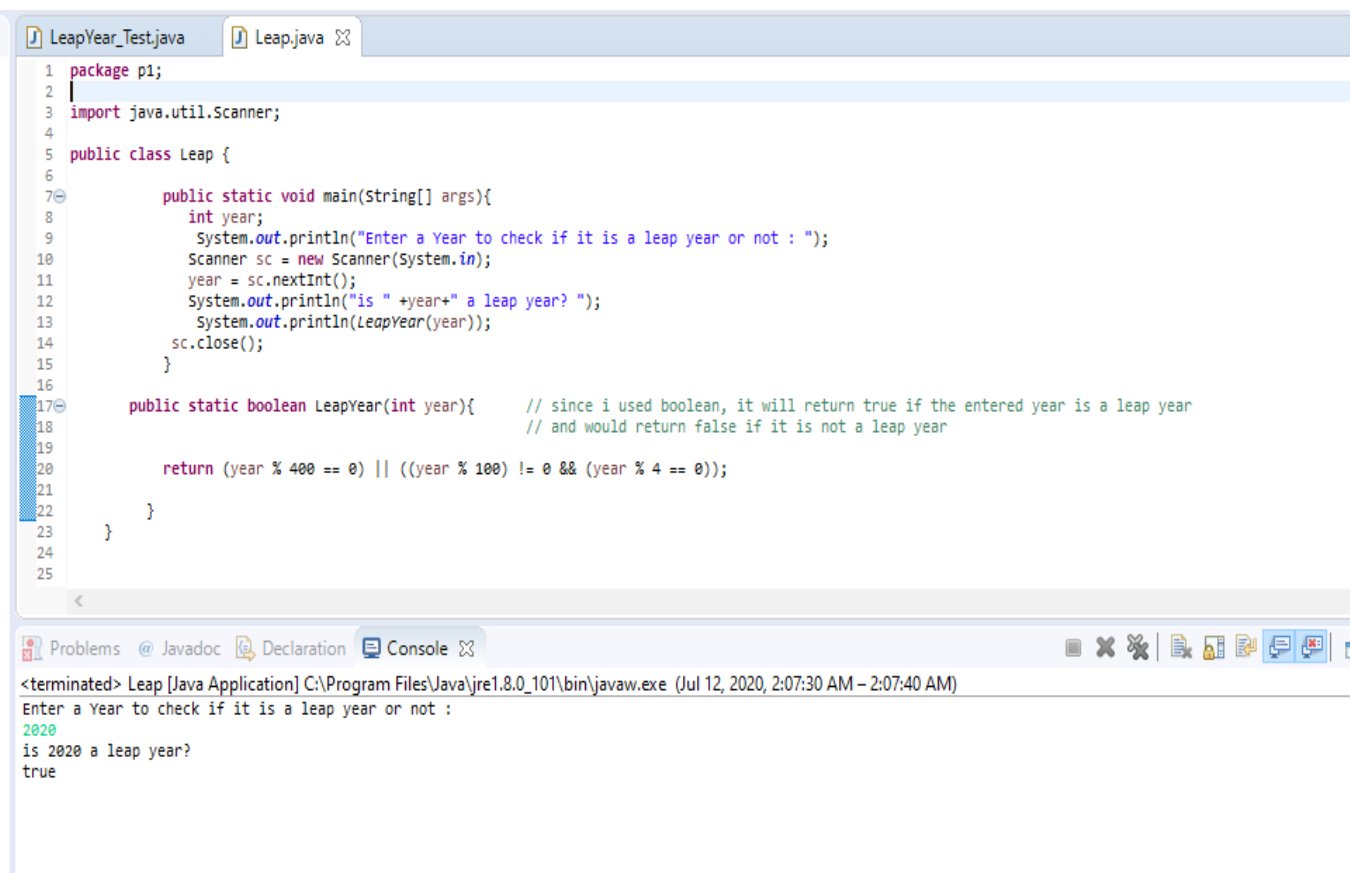
Problems @ Javadoc Declaration Console

```
<terminated> LeapYear_Test [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 12, 2020, 2:04:51 AM - 2:04:53 AM)
Is 2000 a leap year ? : true
Is 2012 a leap year ? : true
Is 1901 a leap year ? : false
Is 1900 a leap year ? : false
```

## Explanation 2:

- In this program I am taking input from the user instead of coding the values myself. I created the class Leap which contains a main method and a static boolean method which only returns the value 'True' or 'False'.
- In the LeapYear method I set some conditions through which the compiler will know how to find a leap year.
- The conditions are that the year that the user entered is divisible by 400 and its remainder equals to zero, it means it is a leap year (the modulo operator). Or, if the year is divisible by 100 and the remainder does not equal to zero and the year is divisible by 4 and the remainder equals to zero. Both of these conditions are the same but just written differently.
- The operator || is called the 'logical or' or simply 'or'. It means as soon as one of the given conditions is true, it will be true.
- The operator && is called the 'and' operator. It is only true when both boolean type values are true.
- It returns true if the entered year is a leap year.

## Program 2 with output:



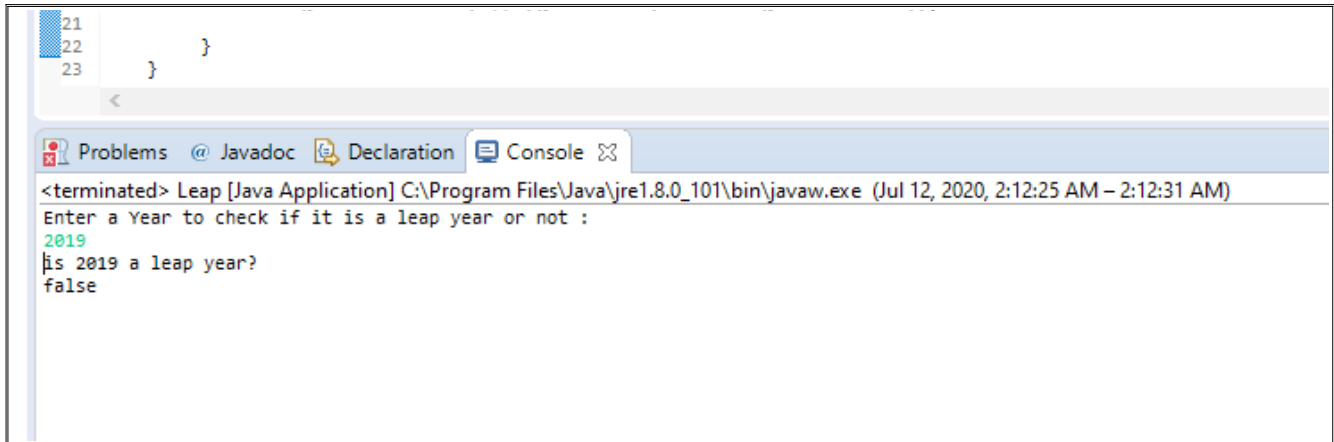
```
LeapYear_Test.java | Leap.java
1 package p1;
2
3 import java.util.Scanner;
4
5 public class Leap {
6
7     public static void main(String[] args){
8         int year;
9         System.out.println("Enter a Year to check if it is a leap year or not : ");
10        Scanner sc = new Scanner(System.in);
11        year = sc.nextInt();
12        System.out.println("is " +year+" a leap year? ");
13        System.out.println(LeapYear(year));
14        sc.close();
15    }
16
17    public static boolean LeapYear(int year){ // since i used boolean, it will return true if the entered year is a leap year
18                                                // and would return false if it is not a leap year
19
20        return (year % 400 == 0) || ((year % 100) != 0 && (year % 4 == 0));
21    }
22 }
23
24
25
```

Problems @ Javadoc Declaration Console

```
<terminated> Leap [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 12, 2020, 2:07:30 AM - 2:07:40 AM)
Enter a Year to check if it is a leap year or not :
2020
is 2020 a leap year?
true
```

- And displays the result as false if the entered year is not a leap year.

### Output 2:



```
21
22     }
23 }

<terminated> Leap [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 12, 2020, 2:12:25 AM – 2:12:31 AM)
Enter a Year to check if it is a leap year or not :
2019
Is 2019 a leap year?
false
```

**Question4.** How to check that the input from the user is the vowel or not in java using object oriented approach?

**Answer:**

**Explanation:**

- The program allows the user to enter an Alphabet thereafter it will check and display the result of the given character whether “it is a vowel” or “consonant” or “not an alphabet” in case the entered character is not an alphabet, using the if else statement.
- As we know that scanner class supports nextInt(), nextDouble(), nextLong(), but there is no nextChar(). So in order to read a char, we use next().charAt(0). The next() function returns the next word in the input as a string, and the char(0) function returns the first character in that string.

## Program:

```
1 package p1;
2
3 import java.util.Scanner;
4
5 class Vowel_or_Consonant {
6
7     void Vowel_Or_Not(char ch)
8     {
9         if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
10        {
11            System.out.println("Entered character "+ch+" is a Vowel");
12        }
13        else if((ch>='a' && ch<='z') || (ch>='A' && ch<='Z'))
14
15            System.out.println("Entered character "+ch+" is Consonant");
16        else
17            System.out.println("Not an alphabet");
18    }
19
20    public static void main(String[] arg)
21    {
22        char ch;
23        Vowel_or_Consonant obj = new Vowel_or_Consonant();
24        Scanner sc = new Scanner(System.in);
25        System.out.println("Enter a character : ");
26        ch = sc.next().charAt(0);
27        obj.Vowel_Or_Not(ch);
28        sc.close();
29    }
30 }
31 }
```

## Output 1:

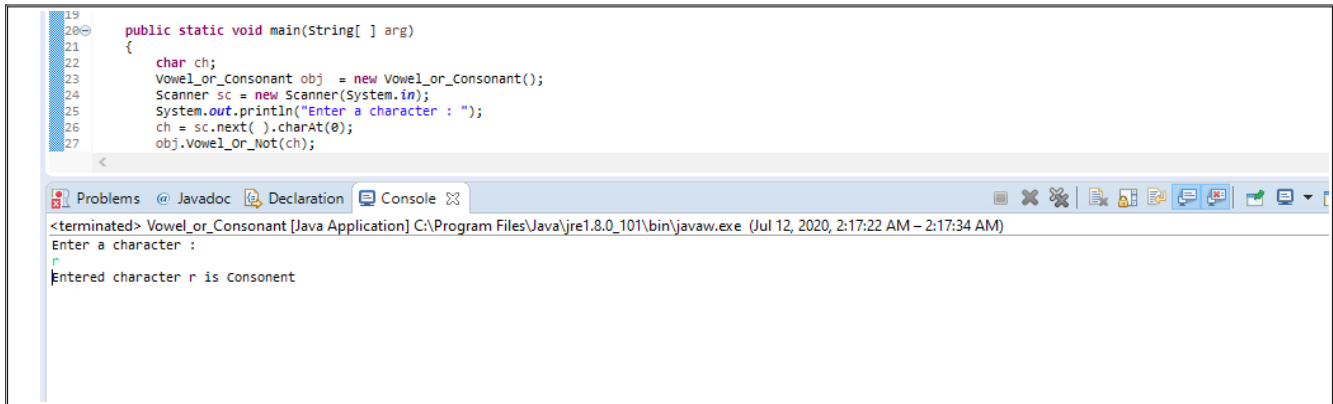
```
18 }
19
20 public static void main(String[] arg)
21 {
22     char ch;
23     Vowel_or_Consonant obj = new Vowel_or_Consonant();
24     Scanner sc = new Scanner(System.in);
25     System.out.println("Enter a character : ");
26     ch = sc.next().charAt(0);
27     obj.Vowel_Or_Not(ch);
28 }
29 }
30 }
31 }
```

<terminated> Vowel\_or\_Consonant [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jul 12, 2020, 2:16:33 AM - 2:16:39 AM)

Enter a character :  
e  
Entered character e is a Vowel



## Output 2:



The screenshot shows an IDE window with a Java code editor and a console. The code in the editor is as follows:

```
19
20 public static void main(String[] arg)
21 {
22     char ch;
23     Vowel_or_Consonant obj = new Vowel_or_Consonant();
24     Scanner sc = new Scanner(System.in);
25     System.out.println("Enter a character : ");
26     ch = sc.next().charAt(0);
27     obj.Vowel_or_Not(ch);
```

The console output shows the program's execution:

```
<terminated> Vowel_or_Consonant [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 12, 2020, 2:17:22 AM - 2:17:34 AM)
Enter a character :
Entered character r is Consonant
```

**Question5.** How to use power of a number in java using object oriented approach?

**Answer:**

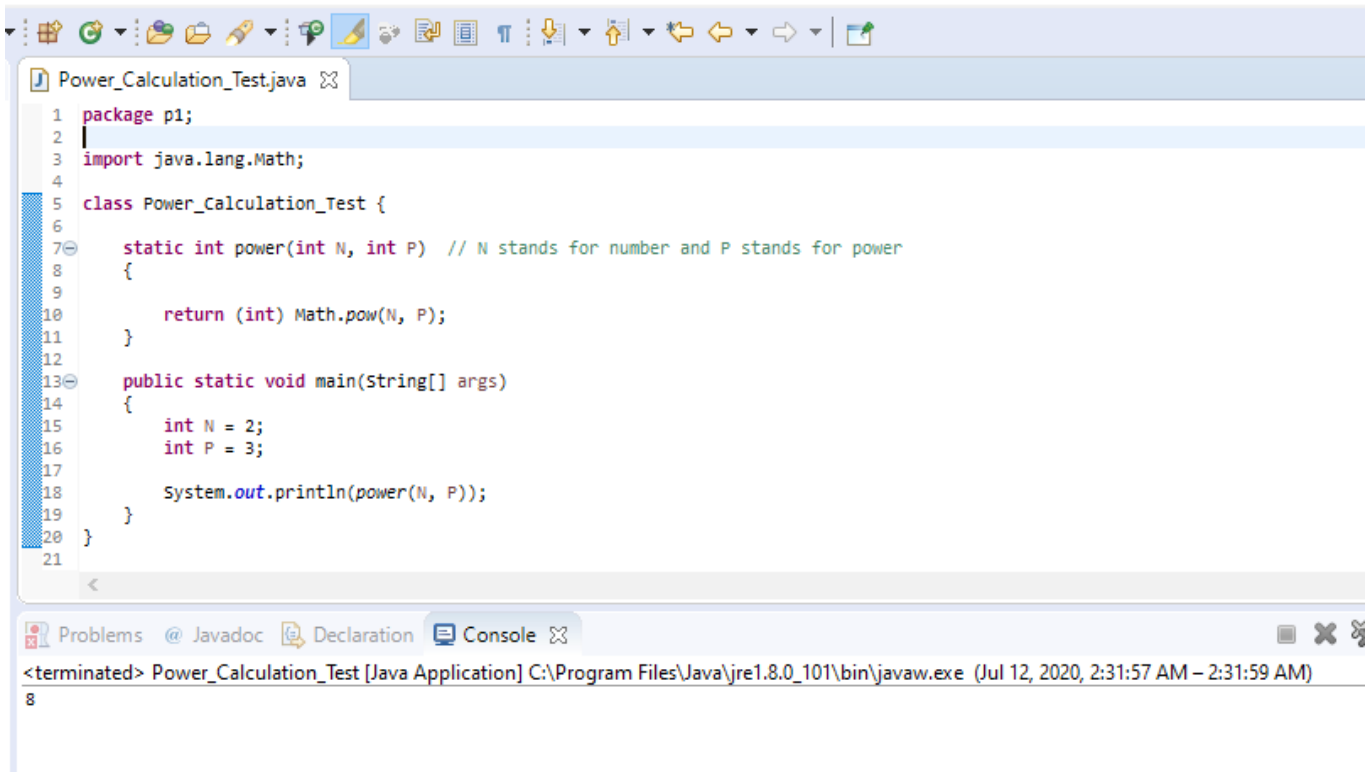
I made two separate and different programs to calculate the power of an integer.

**Explanation 1:**

- In this program I imported the java math class which provides several methods to work on math calculations.
- The static int power method has two parameters, one of them is the integer we will find the power of, and the other one is the value of power we want the integer to raise to.

- The basic math method I used here is `Math.pow()`, which returns the value of the first argument to the power of the second argument.

### Program 1 with output:



The screenshot shows an IDE window titled "Power\_Calculation\_Test.java" with the following code:

```
1 package p1;
2
3 import java.lang.Math;
4
5 class Power_Calculation_Test {
6
7     static int power(int N, int P) // N stands for number and P stands for power
8     {
9
10        return (int) Math.pow(N, P);
11    }
12
13    public static void main(String[] args)
14    {
15        int N = 2;
16        int P = 3;
17
18        System.out.println(power(N, P));
19    }
20 }
21
```

Below the code editor, the "Console" tab is active, showing the output of the program:

```
<terminated> Power_Calculation_Test [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 12, 2020, 2:31:57 AM – 2:31:59 AM)
8
```

### Explanation 2:

- In this program I am using the scanner class to take input from the user.
- The result I was getting was in decimal, so I altered the `Math.pow()` method by writing `(int)` before it when I was calling it, and then I got the output as an integer.

## Program 2 with output:

```
Power_Calculation_Test.java Power_Calculation2.java
1 package p1;
2
3 import java.util.Scanner;
4
5 public class Power_Calculation2 {
6
7     public static void main(String[] args) {
8
9         int a ;
10        int b ;
11
12        System.out.println("enter a number you want to get the power of : " );
13        Scanner s = new Scanner (System.in);
14        a = s.nextInt();
15        System.out.println("enter the power: ");
16        Scanner e = new Scanner (System.in);
17        b = e.nextInt();
18
19        s.close();
20        e.close();
21
22        System.out.println((int)Math.pow(a, b));
23
24    }
25 }
```

Problems @ Javadoc Declaration Console

<terminated> Power\_Calculation2 [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jul 12, 2020, 2:32:53 AM – 2:33:05 AM)

```
enter a number you want to get the power of :
8
enter the power: |
2
64
```