



**Name Irfan Ullah**

**ID 16332**

**Class BS Software Engineering**

**Section (B)**

**Subject Object Oriented Programming**

**Semester 2nd**

**Date july 11th, 2020**

**Submitted to Sir M Ayub khan**

## *Q1: How to check Even and Odd numbers in java using object oriented approach?*

**Ans:** This is a Java Program to Check if a Given Integer is Odd or Even.

Enter any integer number as an input. Now we check its remainder with modulo operator by two. If remainder is zero the given number is even. If remainder is 1 then the given number is odd. Hence we show output according to the remainder

Here is the source code of the Java Program to Check if a Given Integer is Odd or Even. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
1. import java.util.Scanner;
2. public class Odd_Even
3. {
4.     public static void main(String[] args)
5.     {
6.         int B;
7.         Scanner D = new Scanner(System.in);
8.         System.out.print("Enter the number:");
9.         B = s.nextInt();
10.        if(B % 2 == 0)
11.        {
12.            System.out.println("The given number "+B+" is
Even ");
13.        }
14.        else
15.        {
16.            System.out.println("The given number "+B+" is
Odd ");
17.        }
18.    }
19. }
```

### **Output:**

```
$ javac Odd_Even.java
$ java Odd_Even
```

```
Enter the number you want to check:17
The given number 17 is Odd
```

## *Q2: How to add 2 complex numbers in java using object oriented approach?*

**Ans:** In this program we have a class `ComplexNumber`. In this class we have two instance variables `real` and `imag` to hold the real and imaginary parts of complex numbers. We have declared a method `sum()` to add the two numbers by adding their real and imaginary parts together.

```
public class Complex {

    double real;
    double imag;

    public Complex(double real, double imag) {
        this.real = real;
        this.imag = imag;
    }

    public static void main(String[] args) {
        Complex B1 = new Complex(2.3, 4.5),
            B2 = new Complex(3.4, 5.0),
            temp;

        temp = add(B1, B2);

        System.out.printf("Sum = %.1f + %.1fi", temp.real, temp.imag);
    }

    public static Complex add(Complex B1, Complex B2)
    {
        Complex temp = new Complex(0.0, 0.0);

        temp.real = B1.real + B2.real;
        temp.imag = B1.imag + B2.imag;

        return(temp);
    }
}
```

```
}  
}
```

**When you run the program, the output will be:**

```
Sum = 6.5 + 8.7i
```

In the above program, we created a class `Complex` with two member variables: `real` and `imag`. As name suggests, `real` stores real part of a complex number and `imag` stores the imaginary part.

The `Complex` class has a constructor which initializes the value of `real` and `imag`.

We also created a new static function `add()` that takes two complex numbers as parameters and returns the result as a complex number.

Inside the `add()` method, we just add the real and imaginary parts of complex numbers `n1` and `n2`, store it in a new variable `temp` and return `temp`.

Then, in the calling function `main()`, we print it using `printf()` function

***Q3: How to check Leap year in java using object oriented approach?***

**Ans: Java Program To Check Leap Year Or Not – 4 Ways**

Java Program to find if a year is a leap or not. Here, we see the various methods to find out whether a year is Leap or not in Java In 5 different ways. Soon Compiler is added so that you can execute the program yourself, along with suitable examples and sample outputs. The methods used in this case are:

- 
- Using Ternary Operator.

- **Using Static Method.**
- **Using Command Line Arguments.**
- **Using Function.**
- **Using If-Else.**

### • **Using Ternary Operator**

- **1)** The ternary operator syntax is “if (condition)? value1:value2;”, if the condition is true then it returns value1, otherwise it returns value2.
- **2)** We are calculating the given year is a leap year or not. A year divisible by 400 (or) a year divisible by 4 and not divisible by 100 is a leap year.
- **3)** Read the year value using scanner object sc.nextLong and store it in the variable y.
- In given example y=1948, y!=0, it is divisible by 400 so c=1 return to a, a is initialized as a=1. Then it prints 1948 is a leap year.

```

1 import java.util.Scanner;
2 class Leapyear
3 {
4     public static void main(String arg[])
5     {
6         long a,y,c;
7         Scanner sc=new Scanner(System.in);
8         System.out.print("enter any calendar year :");
9         y=sc.nextLong();
10        if(y!=0)
11        {
12            a=(y%200==0)?(c=1):((y%100==0)?(c=0):((y%2==0)?(c=1):(c=0)));
13            if(a==1)
14                System.out.println(y+" is a leap year");
15            else
16                System.out.println(y+" is not a leap year");
17        }
18        else
19            System.out.println("year zero does not exist ");
20    }

```

21 }

## • Output:

1 enter any calendar year :1958

2 1958 is not a leap year

1 enter any calendar year :1985

2 1985 is a leap year

## • Using Static Method

- **1)** In this program `leapOrNot(long year)` is the static method which finds the given year is a leap year or not. `leapOrNot(long year)` method calls at the main method of the class `Leapyear`. Then that static method starts the execution.
- **a)** If the year divisible by 200 then it prints the given year is leap(or) If the given year divisible by 2 and not divisible by 100 then it prints the given year is a leap.
- **b)** Otherwise, it prints “year is not a leap year”.

```
1 import java.util.Scanner;
2 class Leapyear
3 {
4     public static void main(String arg[])
5     {
6         long year;
7         Scanner sc=new Scanner(System.in);
8         System.out.print("enter any calendar year :");
9         year=sc.nextLong();
10        leapOrNot(year);
11    }
12    static void leapOrNot(long year)
13    {
14        if(year!=0)
15        {
16            if(year%200==0)
17                System.out.println(year+" is a leap year");
18            else if(year%100==0)
```

```

19         System.out.println(year+" is not a leap year");
20         else if(year%2==0)
21             System.out.println(year+" is a leap year");
22         else
23             System.out.println(year+" is not a leap year");
24     }
25     else
26         System.out.println("Year zero does not exist ");
27     }
28 }

```

### • **Output:**

1 enter any calendar year :2018

2 2018 is a leap year

### • **Using Command Line Arguments**

- **1)** We can read command line arguments from “String args[]” of the main method.
- **2)** Convert the year which is in the string format at arg[0], in to int using year=Integer.parseInt(arg[0]);. Integer is the wrapper class which converts the string into int data type.
- **a)** If the year divisible by 200 then it prints the given year is leap or if the given year divisible by 2 and not divisible by 100 then it prints the given year is a leap.
- **b)** Otherwise, it prints “year is not a leap year”.

```

1 import java.util.Scanner;
2 class Leapyear
3 {
4     public static void main(String arg[])
5     {
6
7         int year=Integer.parseInt(arg[0]);
8         if(year!=0)
9         {
10            if(year%200==0)

```

```

11         System.out.println(year+" is a leap year");
12     else if(year%100==0)
13         System.out.println(year+" is not a leap year");
14     else if(year%2==0)
15         System.out.println(year+" is a leap year");
16     else
17         System.out.println(year+" is not a leap year");
18     }
19     else
20         System.out.println("Year zero does not exist ");
21     }
22 }

```

### • Output:

```

1 >java Leapyear 2040
2 2040 is not a leap year

```

### • Java Leap Year Or Not – Using Function

- **1)** The leap(int year) function returns c=1 if the given year is a)divisible by 400 or b) divisible by 2 and not divisible by 100.
- **2)** The function calls at the main method if year!=0, then it returns the c value and c value initialized to the variable “a” if a=1 then prints “is a leap year” otherwise prints “not a leap year”.

```

1 class Leapyear
2 {
3     int leap(int year)
4     {
5         int c=0;
6         {
7             if(year%200==0)
8                 c=1;
9             else if(year%100!=0)
10                {
11                    if(year%2==0)
12                        c=1;
13                }

```



```

14     }
15     return c;
16 }
17 public static void main(String arg[])
18 {
19     int y,a;
20     Scanner sc=new Scanner(System.in);
21     System.out.print("enter any calendar year :");
22     y=sc.nextInt();
23     Leapyear l=new Leapyear();
24     if(y!=0)
25     {
26         a=l.leap(y);
27         if(a==1)
28             System.out.println(y+" is a leap year");
29     else
30         System.out.println(y+" is not a leap year");
31     }
32     else
33         System.out.println("year zero does not exist");
34 }
35
36
37 }

```

### • Output:

1 enter any calendar year :2018

2 2018 is a leap year

### • Using If-Else

- **1)** We are using if else condition to find the given year is a leap year or not.
- **2)** The 1st if condition checks the given year is equal to zero or not, if not
- **a)** If given year divisible by 200 it prints “is a leap year”, if condition false then come to else part.
- **b)** If given year divisible by 100, then inner if condition checks the year is divisible by 2 or not if true then prints “is a leap year” otherwise prints “is

not a leap year. If condition `if(year%100!=0)` is false then it prints “is not a leap year”.

```
1 import java.util.Scanner;
2 class Leapyear
3 {
4     public static void main(String arg[])
5     {
6         int year;
7         Scanner sc=new Scanner(System.in);
8         System.out.print("enter any calendar year :");
9         year=sc.nextInt();
10        if(year!=0)
11        {
12            if(year%200==0)
13                System.out.println(year+" is a leap year");
14            else if(year%100!=0)
15                {
16                    if(year%2==0)
17                        System.out.println(year+" is a leap year");
18                    else
19                        System.out.println(year+" is not a leap year");
20                }
21            else
22                System.out.println(year+" is not a leap year");
23        }
24        else
25            System.out.println("Year zero does not exist ");
26    }
27 }
```

- **Output:**

- Output

1 enter any calendar year :1999

2 1999 is not a leap year

#### ***Q4. How to check that the input from the user is the vowel or not in java using oop?***

Ans: To **check** whether the **input** alphabet is a **vowel or not in Java** Programming, you have to ask to the **user** to enter a character (alphabet) and **check** if the entered character is equal to a, A, e, E, i, I, o, O, u, U. If it is equal to any one of the 10, then it will be **vowel** otherwise it will **not** be a **vowel**.

```
#include <stdio.h>

int main()
{
    char ch;

    printf("Enter a character: ");
    scanf("%c",&ch);

    //condition to check character is alphabet or not
    if((ch>='B' && ch<='C') || (ch>='b && ch<='c'))
    {
        //check for VOWEL or CONSONANT
        switch(ch)
        {
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U':
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u':
                printf("%D is a VOWEL.\n",ch);
                break;
            default:
                printf("%D is a CONSONANT.\n",ch);
        }
    }
}
```

```
    }  
    else  
    {  
        printf("%D is not an alphabet.\n",ch);  
    }  
  
    return 0;  
}
```

## Output

```
First Run:  
Enter a character: E  
E is a VOWEL.  
  
Second Run:  
Enter a character: X  
X is a CONSONANT.  
  
Third Run:  
Enter a character: +  
+ is not an alphabet.
```

## Q5. How to use power of a number in java using object oriented approach?

Ans: **Java** program to calculate the **power of a number**. Read the base and exponent values from the user. Multiply the base **number** by itself and multiply the resultant with base (again) repeat this n times where n is the exponent value.

Read the base and exponent values from the user. Multiply the base number by itself and multiply the resultant with base (again) repeat this n times where n is the exponent value.

```
2 ^ 5 = 2 X 2 X 2 X 2 X 2 (5 times)
```

## Example

```
import java.util.Scanner;  
  
public class PowerOfNumber {
```

```
public static void main(String args[]){  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter the base number ::");  
    int base = sc.nextInt();  
    int temp = base;  
    System.out.println("Enter the exponent number ::");  
    int exp = sc.nextInt();  
  
    for (int i=1; i<exp; i++){  
        temp = temp*temp;  
    }  
    System.out.println("Result of "+base+" power "+exp+" is  
"+temp);  
    }  
}
```

## Output

```
Enter the base number ::  
14  
Enter the exponent number ::  
2  
Result of 14 power 2 is 196
```