

**NAME: MUHAMMAD SAEED KHAN**

**ID:16015**

**SUBJECT:OOP**

**DEPARTMENT: BSCS-2**

**QUESTION NUMBER 1: . How to check Even and Odd numbers in java using object oriented approach?**

**ANSWER:**

**Java Program to Check if a Given Integer is Odd or Even**

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.

```
import java.util.Scanner;

public class Odd_Even
{
    public static void main(String[] args)
    {
        int n;
```

```
Scanner s = new Scanner(System.in);

System.out.print("Enter the number you want to check:");

n = s.nextInt();

if(n % 2 == 0)

{

    System.out.println("The given number "+n+" is Even ");

}

else

{

    System.out.println("The given number "+n+" is Odd ");

}

}

}
```

### **OUTPUT**

```
$ javac Odd_Even.java
```

```
$ java Odd_Even
```

```
Enter the number you want to check:7
```

```
The given number 7 is Odd
```



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

### **ANSWER:**

#### **Program to Add Two Complex Numbers**

Given two complex numbers of the form  $a_{\{1\}} + ib_{\{1\}}$  and  $a_{\{2\}} + ib_{\{2\}}$ , the task is to add these two Complex Numbers.

Here the values of real and imaginary numbers is passed while calling the parameterized constructor and with the help of default(empty) constructor, the function addComp is called to get the addition of complex numbers.

#### **EXAMPLE:**

Input:  $a_1 = 4, b_1 = 8$

$a_2 = 5, b_2 = 7$

Output: Sum =  $9 + i15$

Explanation:

$(4 + i8) + (5 + i7)$

$= (4 + 5) + i(8 + 7)$

$= 9 + i15$

Input:  $a_1 = 9, b_1 = 3$

$a_2 = 6, b_2 = 1$

Output: Sum =  $15 + i4$

**The following program is the illustration of the above example**

```
// Java program to Add two complex numbers
```

```
import java.util.*;
```

```
// User Defined Complex class
```

```
class Complex {
```

```
    // Declaring variables
```

```
    int real, imaginary;
```

```
    // Empty Constructor
```

```
    Complex()
```

```
    {
```

```
    }
```

```
    // Constructor to accept
```

```
    // real and imaginary part
```

```
    Complex(int tempReal, int tempImaginary)
```

```
    {
```

```
        real = tempReal;
```

```
        imaginary = tempImaginary;
```

```
}

// Defining addComp() method
// for adding two complex number
Complex addComp(Complex C1, Complex C2)
{
    // creating temporary variable
    Complex temp = new Complex();

    // adding real part of complex numbers
    temp.real = C1.real + C2.real;

    // adding Imaginary part of complex numbers
    temp.imaginary = C1.imaginary + C2.imaginary;

    // returning the sum
    return temp;
}

}

// Main Class
public class GFG {
```

```
// Main function
public static void main(String[] args)
{

    // First Complex number
    Complex C1 = new Complex(3, 2);

    // printing first complex number
    System.out.println("Complex number 1 : "
        + C1.real + " + i"
        + C1.imaginary);

    // Second Complex number
    Complex C2 = new Complex(9, 5);

    // printing second complex number
    System.out.println("Complex number 2 : "
        + C2.real + " + i"
        + C2.imaginary);

    // for Storing the sum
```

```
Complex C3 = new Complex();

// calling addComp() method
C3 = C3.addComp(C1, C2);

// printing the sum
System.out.println("Sum of complex number : "
    + C3.real + " + i"
    + C3.imaginary);
}
}
```

**OUTPUT:**

Complex number 1 : 3 + i2

Complex number 2 : 9 + i5

Sum of complex number : 12 + i7



**QUESTION NUMBER 3: How to check Leap year in java using object oriented approach?**

**ANSWER:**

### **Java Program to Check Leap Year**

In this program, you'll learn to check if the given year is a leap year or not. This is checked using a if else statement.

A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400.

### **Example: Java Program to Check a Leap Year:**

```
public class LeapYear {  
  
    public static void main(String[] args) {  
  
        int year = 1900;  
        boolean leap = false;  
  
        if(year % 4 == 0)  
        {  
            if( year % 100 == 0)  
            {
```



```
// year is divisible by 400, hence the year is a leap year
if ( year % 400 == 0)
    leap = true;
else
    leap = false;
}
else
    leap = true;
}
else
    leap = false;

if(leap)
    System.out.println(year + " is a leap year.");
else
    System.out.println(year + " is not a leap year.");
}
}
```

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

1900 is not a leap year.

**When you change the value of year to 2012, the output will be:**

2012 is a leap year.

In the above program, given year 1900 is stored in the variable year.

Since 1900 is divisible by 4 and is also a century year (ending with 00), it has to be divisible by 400 for a leap year. Since it's not divisible by 400, 1900 is not a leap year.

But, if we change year to 2000, it is divisible by 4, is a century year and is also divisible by 400. So, 2000 is a leap year.

Likewise, If we change year to 2012, it is divisible by 4 and is not a century year, so 2012 is a leap year. We don't need to check if 2012 is divisible by 400 or not.



**QUESTION NUMBER 4: . How to check that the input from the user is the vowel or not in java using object oriented approach?**

**ANSWER:**

### **Java Program to Check Vowel or Not**

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.

### **Java Programming Code to Check Vowel or Not:**

```
/* Java Program Example - Check for Vowel */
```

```
import java.util.Scanner;
```

```
public class JavaProgram
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        char ch;
```

```
        Scanner scan = new Scanner(System.in);
```

```
        System.out.print("Enter an Alphabet : ");
```

```
        ch = scan.next().charAt(0);
```

```
if(ch=='a' || ch=='A' || ch=='e' || ch=='E' ||
ch=='i' || ch=='I' || ch=='o' || ch=='O' ||
ch=='u' || ch=='U')
{
    System.out.print("This is a Vowel");
}
else
{
    System.out.print("This is not a Vowel");
}
}
```

When the above Java Program is compile and executed, it will produce the following result. Above Java Programming Example Output (for vowel):

**OUTPUT:**

Enter an Alphabet : e

This is a vowel

**Example Output (for not vowel):**

Enter an Alphabet : t

This is not a vowel.



## QUESTION NUMBER 5: How to use power of a number in java using object oriented approach?

### ANSWER:

#### Java Program to Calculate Power of a Number:

In this program we are calculating the power of a given number using for loop. Here number is the base and p is the power (exponent). So we are calculating the result of  $\text{number}^p$ .

```
public class JavaExample {  
  
    public static void main(String[] args) {  
  
        //Here number is the base and p is the exponent  
  
        int number = 2, p = 5;  
  
        long result = 1;  
  
  
        //Copying the exponent value to the loop counter  
  
        int i = p;  
  
        for (;i != 0; --i)  
  
        {  
  
            result *= number;  
  
        }  
  
  
        //Displaying the output  
  
        System.out.println(number+"^"+p+" = "+result);  
    }  
}
```

```
}
```

```
}
```

**OUTPUT:**

$2^5=32$

