**Department of computer Science**

**Assignment**

**Name : Laden Abbas**

**ID: 16268**

**Subject: Object Oriented Programming (Lab)**

**Instructor: M.Ayub Khan**

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

**Ans:**

This is a Java Program to Print the Odd & Even Numbers in an Array.

Enter size of array and then enter all the elements of that array. Now using for loop and if codition we use to distinguish whether given integer in the array is odd or even.

Here is the source code of the Java Program to Print the Odd & Even Numbers in an Array. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

import java.util.Scanner;

public class Even\_Odd

{

 public static void main(String[] args)

 {

 int n;

 Scanner s = new Scanner(System.in);

 System.out.print("Enter no. of elements you want in array:");

 n = s.nextInt();

 int a[] = new int[n];

 System.out.println("Enter all the elements:");

 for (int i = 0; i < n; i++)

 {

 a[i] = s.nextInt();

 }

 System.out.print("Odd numbers:");

 for(int i = 0 ; i < n ; i++)

 {

 if(a[i] % 2 != 0)

 {

 System.out.print(a[i]+" ");

 }

 }

 System.out.println("");

 System.out.print("Even numbers:");

 for(int i = 0 ; i < n ; i++)

 {

 if(a[i] % 2 == 0)

 {

 System.out.print(a[i]+" ");

 }

 }

 }

}

Output:

$ javac Even\_Odd.java

$ java Even\_Odd

Enter no. of elements you want in array:5

Enter all the elements:

1

2

3

4

5

Odd numbers:1 3 5

Even numbers:2 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**Ans:**

Java Program to Add Two Complex Numbers

BY CHAITANYA SINGH | FILED UNDER: JAVA EXAMPLES

Complex numbers have two parts – real part and imaginary part. In this tutorial, we will write a Java program to add two complex numbers. When adding complex numbers we add real parts together and imaginary parts together as shown in the following diagram.

Java Add two complex numbers

Example – Adding two complex numbers in Java

In this program we have a class ComplexNumber. In this class we have two instance variables real and img 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.

The constructor of this class is used for initializing the complex numbers. For e.g. when we create an instance of this class like this ComplexNumber temp = new ComplexNumber(0, 0);, it actually creates a complex number 0 + 0i.

public class ComplexNumber{

 //for real and imaginary parts of complex numbers

 double real, img;

 //constructor to initialize the complex number

 ComplexNumber(double r, double i){

 this.real = r;

 this.img = i;

 }

 public static ComplexNumber sum(ComplexNumber c1, ComplexNumber c2)

 {

 //creating a temporary complex number to hold the sum of two numbers

 ComplexNumber temp = new ComplexNumber(0, 0);

 temp.real = c1.real + c2.real;

 temp.img = c1.img + c2.img;

 //returning the output complex number

 return temp;

 }

 public static void main(String args[]) {

 ComplexNumber c1 = new ComplexNumber(5.5, 4);

 ComplexNumber c2 = new ComplexNumber(1.2, 3.5);

 ComplexNumber temp = sum(c1, c2);

 System.out.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");

 }

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**Ans:**

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 divisble by 4 and is also a century year (ending with 00), it has be divisble 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 a leap year. We don't need to check if 2012 is divisible by 400 or not.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q4. How to check that the input from the user is the vowel or not in java using object oriented approach?**

**Ans:**

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

Following Java Program ask to the user to enter an alphabet to check whether it is a vowel or not, then display the result on the screen:

/\* 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):

Java Program check vowel

Above Java Programming Example Output (for not vowel):

check vowel in java

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**Ans:**

Java Program to Calculate the Power of a Number

In this program, you'll learn to calculate the power of a number with and without using pow() function.

Example 1: Calculate power of a number using a while loop

public class Power {

 public static void main(String[] args) {

 int base = 3, exponent = 4;

 long result = 1;

 while (exponent != 0)

 {

 result \*= base;

 --exponent;

 }

 System.out.println("Answer = " + result);

 }

}

When you run the program, the output will be:

Answer = 81

In this program, base and exponent are assigned values 3 and 4 respectively.

Using the while loop, we keep on multiplying result by base until exponent becomes zero.

In this case, we multiply result by base 4 times in total, so result = 1 \* 3 \* 3 \* 3 \* 3 = 81.

Example 2: Calculate power of a number using a for loop

public class Power {

 public static void main(String[] args) {

 int base = 3, exponent = 4;

 long result = 1;

 for (;exponent != 0; --exponent)

 {

 result \*= base;

 }

 System.out.println("Answer = " + result);

 }

}

When you run the program, the output will be:

Answer = 81