

Name----- Hidayatullah

ID----- 16495

Section----- (B)

Department----- BS Software Engineering

Semester----- 2nd

Subject----- Object Oriented Programming

Examination----- Final-Term

Date----- July 12th, 2020

Teacher----- M.Ayub Khan

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

Answer: **Program**

```
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
```

Explanation:

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.



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

Answer: **Program**

```
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
```

Explanation:

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.

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 with 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?

Answer: 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.

```
import java.util.Scanner;  
class Leapyear
```

```
{
    public static void main(String arg[])
    {
        long a,y,c;
        Scanner sc=new Scanner(System.in);
        System.out.print("enter any calendar year :");
        y=sc.nextLong();
        if(y!=0)
        {
            a=(y%200==0)?(c=1):((y%100==0)?(c=0):((y%2==0)?(c=1):(c=0)));
            if(a==1)
                System.out.println(y+" is a leap year");
            else
                System.out.println(y+" is not a leap
year");
        }
        else
            System.out.println("year zero does not exist
");
    }
}
```

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
```



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?

Answer: 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.

$2^5 = 2 \times 2 \times 2 \times 2 \times 2$ (5 times)

Program

```
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
```



Thank You My Respectable Teacher...!