|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Object Oriented Programming** | **Semester:** | **III** |
| **Date of Performance:** | **24 / 07 / 2023** | **Batch No:** | **A3** |
| **Faculty Name:** | **Prof. Pragya Gupta** | **Roll No:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_/25** |

|  |  |  |
| --- | --- | --- |
| **Writing Program (07)** | **Performance in lab**  **and Viva (05+03)** | **Post lab questions, conclusion and completion (03+02+05)** |
|  |  |  |

**Experiment No: 1**

**Title: Basic Java Program with use of Operators and Control**

**Flow**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Learn basic Java Program with use of Operators and Control Flow |

|  |
| --- |
| **COs to be achieved:** |
| **CO1**: Understand concepts of Object Oriented Programming and basic characteristics of Java. |

|  |
| --- |
| **Tools used:** |
|  |

|  |
| --- |
| **Theory:** |
| **(Mention about data type, operator and flow statement)**  **Java Scanner Class**  Java **Scanner class** allows the user to take input from the console. It belongs to **java.util** package. It is used to read the input of primitive types like int, double, long, short, float, and byte. It is the easiest way to read input in Java program.  Syntax   1. Scanner sc=**new** Scanner(System.in);   The above statement creates a constructor of the Scanner class having **System.inM** as an argument. It means it is going to read from the standard input stream of the program. The **java.util** package should be import while using Scanner class.  It also converts the Bytes (from the input stream) into characters using the platform's default charset.     1. Scanner sc= **new** Scanner(System.in);    //System.in is a standard input stream 2. System.out.print("Enter first number - "); 3. **int** a= sc.nextInt(); |

|  |
| --- |
| **Code:** |
| 1. Find the perimeter and area of circle. Input will be entered by user (You should use Math.PI constant in your program) If input is not valid then print appropriate message.   import java.util.Scanner;  class circle {      public static void main(String[] args)      {          float radius, area, peri;          Scanner sc = new Scanner(System.in);          System.out.println("\nenter the radius of the circle: ");          radius = sc.nextFloat();          if (radius > 0)          {              System.out.println("input is VALID");              peri = 2 \* (float) Math.PI \* radius;              System.out.println("\nperimeter = " + peri);              area = (float) (Math.PI \* radius \* radius);              System.out.println("area = " + area);          }            else          {              System.out.println("INVALID radius entered");          }        }  }   1. Consider first **n even numbers** starting from zero (0). Write a program to calculate sum of all the numbers divisible by 3 from 0 to n. Print the sum. If input is not valid then print appropriate message.   import java.util.Scanner;  class exp1\_q2  {      public static void main(String[] args)      {          int num, sum = 0;          Scanner sc = new Scanner(System.in);    System.out.println("\nenter a number: ");          num = sc.nextInt();          for(int i=0; i<=num; i++)          {              if (i%2==0 && i%3==0)              {                  sum = sum + i;                  i++;              }          }          System.out.println("sum of even numbers divisible by 3 are: " + sum);      }  } |

|  |
| --- |
| **Output:** |
|  |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. **Write a program to find the largest among three numbers x,y, and z. You should use if-then-else construct in Java**.   import java.util.Scanner;  public class exp1\_plq1  {      public static void main(String[] args)      {          int largest;          Scanner sc = new Scanner(System.in);          System.out.println("enter the first number (x): ");              int x = sc.nextInt();          System.out.println("enter the second number (y): ");              int y = sc.nextInt();          System.out.println("enter the third number (z): ");              int z = sc.nextInt();          if (x>=y && x>=z)          {              largest = x;          }            else if (y>=x && y>=z)          {              largest = y;          }            else          {              largest = z;          }          System.out.println("largest number is: " + largest);      }  }     1. Write a program to check whether the number is an Armstrong number or not.   import java.util.\*;  class exp1\_plq2  {    public static void main(String[] args)    {      int digit, result = 0;      Scanner sc = new Scanner(System.in);        System.out.print("\nenter a number: ");      int num = sc.nextInt();      int originalNum = num;      while(num>0)      {        digit = num%10;        num = num/10;        result = result + (digit \* digit \* digit);      }      if(originalNum == result)          System.out.println("amrstrong number");        else          System.out.println("NOT an armstrong number");    }  }  // armstrong number (153 = 1\*1\*1+5\*5\*5+3\*3\*3)     1. **Explain the difference between class and object.**   Class is a detailed description, the definition, and the template of what an object will be. But it is not the object itself. Also, what we call, a class is the building block that leads to Object-Oriented Programming. It is a user-defined data type, that holds its own data members and member functions, which can be accessed and used by creating an instance of that class. It is the blueprint of any object. Once we have written a class and defined it, we can use it to create as many objects based on that class as we want.  In Java, the class contains fields, constructors, and methods. For example, consider the Class of Accounts. There may be many accounts with different names and types, but all of them will share some common properties, as all of them will have some common attributes like balance, account holder name, etc. So here, the Account is the class.  Object is an instance of a class. All data members and member functions of the class can be accessed with the help of objects. When a class is defined, no memory is allocated, but memory is allocated when it is instantiated (i.e., an object is created).  For Example, considering the objects for the class Account are SBI Account, ICICI account, etc.   1. **Why is java known as platform independent language?**   Java is platform-independent because it uses a virtual machine. The Java programming language and all APIs are compiled into bytecodes. Bytecodes are effectively platform-independent. The virtual machine takes care of the differences between the bytecodes for the different platforms. The run-time requirements for Java are therefore very small. The Java virtual machine takes care of all hardware-related issues so that no code has to be compiled for different hardware. |

|  |
| --- |
| **Conclusion:** |
| Successfully learnt how to write the basic Java programs with the use of operators and control flow. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |