



Assignment No 1

- I. Explain what is platform dependency & what is platform independency ?
→ Platform dependency :-

Platform dependency in Java refers to the characteristics of Java programs that allows them to run on any platform that has a Java Virtual Machine (JVM) installed, without needing to be recompiled.

Platform independency :-

Platform independence in Java, often referred to as "write once, run anywhere" means that Java programs can be developed on one platform & executed on another platform without modification.

2. Explain how data entered using command line arguments with example.
→ In Java, data can be entered via the command line, using the 'args' parameter of the 'main' method. When you run a Java program from the command line, you can provide input arguments that are



passed to the program through the 'args' parameter.

Example :-

```
class Demol
{
    public static void main (String [] args)
    {
        if (args.length > 0)
        {
            for (int i=0; i<args.length; i++)
            {
                System.out.println ("Argument " + (i+1) + " "
                    + args[i]);
            }
        }
        else
        {
            System.out.println ("No command line
                arguments provided.");
        }
    }
}
```

Output :-

Argument 1: hello
Argument 2: world

3. Explain how unary operator in java with example.

→ Java unary operator are the types that need only one operand to perform any operation like increment, decrement, negation etc

- It consist of various arithmetic, logical & other operator that operate on a single operator.

Unary (-)

Not operator (!)

Increment (++)

Decrement (--)

Bitwise complement (~)

Eg. import java.util.Scanner;

class Demounary

```
{  
    public static void main (String [] args)  
    {  
        int x = 0;  
    }  
}
```

```

int y=0,
int z=0.
Scanner sc = new Scanner (System.in);
System.out.println ("Enter the value
of x & y:");
x = sc.nextInt ();
y = sc.nextInt ();
z = -x;
x++;
--y;
System.out.println ("x = " + x + " y = " + y
+ " z = " + z);
System.out.println ("x = " + x + " y = " + y +
+ " z = " + z);
}

```

- q4 Write a c program to input marks of FIVE subject physics ,chemistry ,biology ,mathematics & computer calculate % percentage $\geq 90\%$: percentage Grade A
percentage $\geq 80\%$: Grade B
percentage $\geq 70\%$: Grade C
percentage $\geq 60\%$: Grade D
percentage $\geq 40\%$: Grade E
percentage $\leq 40\%$: Grade F



```

#include <stdio.h>
int main ()
{
float physics , chemistry , biology , mathematics
computer ,
float total_marks , percentage ;
printf ("Enter marks of physics :"),
scanf ("%f" , &physics),
printf ("Enter marks of chemistry :"),
scanf ("%f" , &chemistry),
printf ("Enter the marks of Biology :"),
scanf ("%f" , &Biology),
printf ("Enter the marks of Mathematics :"),
scanf ("%f" , &Mathematics),
printf ("Enter the marks of computer :"),
scanf ("%f" , &computer),
total_marks = physics + chemistry + biology +
mathematics + computer ;
percentage = (total_marks / 500) * 100 ;
char grade ;
if (percentage  $\geq 90$ )
{
    grade = 'A' ;
}
else if (percentage  $\geq 80$ )
{
    grade = 'B' ;
}

```

```

else if (percentage >= 20)
{
    grade = 'C';
}
else if (percentage >= 60)
{
    grade = 'D';
}
else if (percentage >= 40)
{
    grade = 'E';
}
else
{
    grade = 'F';
}
printf("percentage : %.2f\n", percentage);
printf("Grade : %c\n", grade);
return 0;
}

```

Q 5

Write a program to enter the runs scored by a player in three matches & display the highest scored using ?: operator?

```

import java.util.Scanner;
public class Main
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter runs scored in match 1:");
        int match1 = sc.nextInt();
        System.out.println ("Enter runs scored in match 2:");
        int match2 = sc.nextInt();
        System.out.println ("Enter runs scored in match 3:");
        int match3 = sc.nextInt();
        int highestScore = (match1 > match2) ?
            (match1 > match3) ? match1 : match3 :
            (match2 > match3) ? match2 : match3;
        System.out.println ("The highest score is " +
            highestScore);
        scanner.close();
    }
}

```

6] Write a program to check number is Abundant number or not?

```

→ import java.util.Scanner;
class AbundantNo
{
    public static void main (String []args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter a number :");
        int number = sc.nextInt ();
        if (isAbundant (number))
        {
            System.out.println ("number + " is an
                abundant number ");
        }
        else
        {
            System.out.println ("number + " is an
                not an abundant number ");
        }
        scanner.close ();
    }

    static boolean isAbundant (int n)
    {
        if (n <= 1)
            return false;
    }
}

```

```
int sum=1;
for (int i=2, i*x <=n ; i++)
{
    if (n % i == 0)
        if (i * i != n)
            sum = sum + n / i;
    else
        sum += i;
}
return sum > n;
```

7] Write a program to calculate the factorial of entered number.
→ import java.util.Scanner;
class Factorial
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);

```

Dr. D. Y. PATIL
EDUCATIONAL INSTITUTIONS
system.out.println("Enter a number");
int number = sc.nextInt();
long factorial = calculatefactorial(number);
System.out.println("Factorial of " + number
    " is " + factorial);
sc.close();
}

static long calculatefactorial (int n)
{
    if (n == 0 || n == 1)
    {
        return 1;
    }
    else
    {
        return n * calculatefactorial (n-1);
    }
}

```

8) Write a program to print following pattern.

```

* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *

```

class starpattern

```

public static void main (String []args)
{
    int rows = 10;
    for (int i=0 ; i < rows ; i++)
    {
        for (int j=0 ; j < i ; j++)
        {
            System.out.print(" ");
        }
        System.out.print (" * ");
        System.out.print (" ");
    }
}

```

b] A

```

A B A
A B C B A
B C D C B A
B C D E D C B A

```

→ class pattern

```

public static void main (String []args)
{
}

```

```

{
    for (int i = 2, i < limit, i++)
    {
        int nextNum = num1 + num2;
        System.out.println(nextNum + " ");
        num1 = num2;
        num2 = nextNum;
    }
}

```

Q] Write a program to print fibonacci series given ahead 0 1 1 2 3 5 8 13
21 34

→ class fibonacciseries

```

{
    public static void main (String [] args)
    {
        int limit = 13;
        int num1 = 0;
        System.out.print (num1 + " " + num2 + " ");
        for (int i = 2, i < limit, i++)
        {
            int nextNum = num1 + num2;
            System.out.print (nextNum + " ");
            num1 = num2;
            num2 = nextNum;
        }
    }
}

```

num1 = num2;
num2 = nextNum;
3
3

10] Write a program /menu driven program to perform different arithmetic operations using switch case.

→ import java.util.Scanner;
class menuDriven

```

public static void main (String [] args)
{
    Scanner sc = new Scanner (System.in);
    int a, b, ab;
    System.out.println ("1. Addition");
    System.out.println ("2. Subtraction");
    System.out.println ("3. Multiplication");
    System.out.println ("4. Division");
    System.out.println ("Select your choice  
1-4");
    switch (ch)
    {
        case 1:
            System.out.println ("Enter the number");
            a = input.nextInt();
    }
}

```

```

ODD V PATH
system.out.println("Enter the number 1:");
b = input.nextInt();
system.out.println("Addition = " + (a+b));
break;

case 2:
system.out.println("Enter the number 1:");
a = input.nextInt();
system.out.println("Enter the number 2:");
b = input.nextInt();
system.out.println("Subtraction = " + (a-b));
break;

case 3:
system.out.println("Enter the number 1:");
a = input.nextInt();
system.out.println("Enter the number 2:");
b = input.nextInt();
system.out.println("Multiplication = " + (a*b));
break;

case 4:
system.out.println("Enter the number 1:");
a = input.nextInt();
system.out.println("Enter the number 2:");
b = input.nextInt();
break;

```

ODD V PATH

default
 system.out.println("invalid choice
 ,
 , please try again !!!"),
 }
 }

11. Write a program to sort an array
 descending order
 → import java.util.Scanner;
 import java.util.Collections;
 class DescendingSort
 {
 public static void main(String args)
 {
 Integer [] array = {5, 2, 8, 1, 3};
 array.sort(Collections.reverseOrder());
 system.out.println("sorted array in
 descending order");
 for (int sum : array)
 {
 system.out.println(num + " ");
 }
 }
 }



Assignment No. 2

1. Explain how platform independency achieved in Java?

→ 1. compilation to byte code :- Java source code is compiled into bytecode which is platform independent intermediate representation.

2. Java virtual machine (JVM) :- The JVM is an integral part of java program it interprets the bytecode & translate it into machine code for the specific hardware operating system is running on.

3. standard library (Java API) :- Java provides is a comprehensive standard library that abstracts away many low level details of underlying os.

4. No Native code :- This eliminates platform specific dependencies that run arise from using native libraries.

- overall these mechanism work together to ensure that java programs can be written once & run anywhere making

Java truly platform independent lang

a 2] Write a program using BufferedReader & InputStreamReader to input & display the elements of different types & explain it.

```
→ import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
class InputDisplayElements
{
    public static void main (String [] args)
    BufferedReader reader = new BufferedReader(new
        InputStreamReader (System.in));
    try
    {
        System.out.println ("Enter an integer");
        int integerInput = Integer.parseInt
            (reader.readLine ());
        System.out.print ("Enter a double :");
        double doubleInput = Double.parseDouble
            (reader.readLine ());
        System.out.println ("you entered a double: "
            + doubleInput );
        System.out.print ("Enter a string :");
        String stringInput = reader.readLine (),
    }
```

System.out.println ("you entered a
 string " + stringInput),
 }
 catch (IOException)
 {
 System.out.println ("Error reading input"
 + e.getMessage ());
 }
 catch (NumberFormatException e)
 {
 System.out.println ("Invalid input format"
 + e.getMessage ());
 }
 finally
 {
 try
 {
 reader.close ();
 }
 catch (IOException)
 {
 System.out.println ("Error closing Reader"
 + e.getMessage ());
 }
 }
}



Dr. D. Y. PATIL
EDUCATIONAL FEDERATION

- 3] Explain arithmetic operators in java with example.
- 1] Arithmetic (+) :- Add two operands
eg int result = 5 + 3 ; = 8
- 2] Subtraction (-) :- subtract the second operand from the first
int result = 7 - 2 ; = 5
- 3] Multiplication (*) :- Multiplies two operand
int result = 4 * 6 ; =
- 4] Division (/) :- divides first operand by second double result
double result = 10 / 3 ;
- increment (++) :- increase the value of the operand by 1
int a = 5 ;
a++ ;
- Decrement (--) :- decrease the value of operand by 1
int b = 8
b-- ;

- 4] Write a c program to input basic salary of employees & calculate its gross salary according to the following ?
- Basic salary <= 1000 : HRA = 20% DA = 80%
- Basic salary <= 2000 : HRA = 25% DA = 90%
- Basic salary >= 2000 : HRA = 30% DA = 95%
- #include <stdio.h>
- ```
int main () {
 float basic, salary, gross_salary, hra, da;
 printf ("Enter the basic salary of an employee : ");
 scanf ("%f", &basic_salary);
 if (basic_salary <= 1000)
 hra = 0.2 * basic_salary;
 da = 0.8 * basic_salary;
 }
 else
 {
 hra = 0.3 * basic_salary;
 da = 0.95 * basic_salary;
 }
 gross_salary = basic_salary + hra + da;
 printf ("Gross salary : %.2f\n", gross_salary);
 return 0;
}
```



Q8) Write a program to find the entered number is divisible by 3 & 5 or not using %

```

→ import java.util.Scanner
public class DivisibleByThreeAndFive
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 int number = sc.nextInt ();
 if (number % 3 == 0 && number % 5 == 0)
 System.out.println ("Number + " is divisible
 by both 3 & 5");
 else
 System.out.println ("Number + " is not
 divisible by both 3 & 5");
 sc.close ();
 }
}

```

Write a program to check entered number is Automorphic number or not?

→

```

import java.util.Scanner;
public class AutomorphicNumber
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.println ("Enter a number");
 int number = sc.nextInt ();
 if (isAutomorphic (number + " is an automorphic number"));
 else
 System.out.println (number + " is not
 automorphic number");
 }

 public static boolean isAutomorphic
 (int number)
 {
 int square = number * number;
 String numbersstr = String.valueOf (number);
 String squarestr = String.valueOf (square);
 return squarestr.endsWith (numbersstr);
 }
}

```



7] Write a program to calculate HCF (GCD) of two numbers

```
→ class HCF
{
 public static void main (String [] args)
 {
 int num 1 = 24, num 2 = 36, hcf = 0;
 for (int i = 1; i <= num 1 || i <= num 2; i++)
 if (num 1 % i == 0 && num 2 % i == 0)
 hcf = i;
 System.out.println ("The HCF : " + hcf);
 }
}
```

8] Write a program to check whether either enter number is wnsdence no or not.

```
import java.util.Scanner;
class condenseNumber
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.println ("Enter a Number :");
 int number = scanner.nextInt();
 scanner.close();
 }
}
```

if (isWnsdence (number))
{
 System.out.println (number + " is a condense number :");
}
else
{
 System.out.println (number + " is not a condense number :");
}
public static boolean isCondense
 (int number)
{
 String strNumber = Integer.toString
 (number);
 for (int i = 0, i < strNumber.length () - 1; i++)
 int digit 1 = Character.getNumericValue
 (strNumber.charAt (i));
 int product = digit \* digit 2;
 if (product != 0 && number % product == 0)
 return true;
}



Dr. D. Y. PATIL  
EDUCATIONAL FOUNDATION

return false;

g] Write a program in java to find sum of given series  $s = 1! + 2! + 3! + \dots + n!$

```
→ import java.util.Scanner;
public static void main (String [] args)
{
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter the value of n:");
 int n = sc.nextInt ();
 sc.close ();
 long sum = sumOfFactorials (n);
 System.out.println ("sum of the series up to " + n + " terms : " + sum);
}

public static long sumOfFactorials (int n)
{
 long sum = 0;
 long factorial = 1;
 for (int i=1; i<=n; i++)
 {
 factorial *= i;
 sum += factorial;
 }
}
```

10. Write a menu driven program to perform prime check reverse Armstrong palindrome using switch case

```
→ import java.util.Scanner;
class MenuDrivenProgram
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 int choice =
 do
 {
 System.out.println ("Menu");
 System.out.println ("1. prime check");
 System.out.println ("2. reverse a number");
 System.out.println ("3. Armstrong check");
 System.out.println ("4. palindrome check");
 System.out.println ("5. Exit");
 System.out.println ("Enter your choice:");
 choice = sc.nextInt ();
 switch (choice)
 {
 case 1:
 primecheck ();
 break;
 }
 } while (choice != 5);
 }
}
```



Dr. D. Y. PATIL  
SCHOOL OF ENGINEERING

```
case 2:
 reverseNumber();
 break;
case 3:
 armstrongCheck();
 break;
case 4:
 palindromeCheck();
 break;
case 5:
 System.out.println("Invalid choice...
 please enter a number betw 1 & 5");
default:
 while (choice != 5);
 sc.close();
public static void primecheck()
{
 Scanner sc = new Scanner (System.in);
 System.out.print("Enter a number : ");
 check if its prime ");
 int number = sc.nextInt();
 sc.close();
 boolean isprime = true;
 if (number <= 1)
 {
```

isprime = false;  
}  
else  
{  
 for (int i=2, i <= math.sqrt (number),  
 i++)  
 {  
 if (number % i == 0)  
 {  
 isprime = false;  
 break;  
 }  
 }  
 if (isprime)  
 {  
 System.out.println (number + " is a prime  
 no.");  
 }  
 else  
{  
 System.out.println (number + " is a not  
 prime number.");  
 }  
}  
public static void reverseNumber()  
{

```

Dr. D. Y. PATIL
EDUCATIONAL FOUNDATION

Scanner sc = new Scanner (System.in);
System.out.println ("Enter a number to
reverse :");
int number = sc.nextInt ();
sc.close ();
int reverseNumber = 0;
while (number != 0)
{
 int digit = number % 10;
 reverseNumber = reverseNumber * 10 + digit;
 number /= 10;
}
System.out.println ("Reverse number :" +
reverseNumber);
}

public static void armstrongcheck ()
{
Scanner sc = new Scanner (System.in);
System.out.print ("Enter a number to
check if its Armstrong :");
int number = sc.nextInt ();
sc.close ();
int originalNumber = Number;
int numberOfDigits = String.valueOf (no).
length ();
int sum = 0;
while (number != 0)

```

```

Dr. D. Y. PATIL
EDUCATIONAL FOUNDATION

int digit = number % 10;
sum += Math.pow (digit, Number.digits);
number /= 10;
if (sum == originalNumber)
{
 System.out.println ("Original Number " +
 "is
an Armstrong Number");
}
else
{
 System.out.println ("Original Number " +
 "is not an Armstrong number :");
}
}

public static void palindromecheck ()
{
Scanner sc = new Scanner (System.in);
System.out.print ("Enter a number to
check if its palindrome :");
int number = Scanner.nextInt ();
Scanner.close ();
int originalNumber = number;
int reverseNumber = 0;
while (number != 0)
{
 int digit = number % 10;

```

```

reverseNumber = reverse number * 10 ^ digits,
Number != 10;
if (originalNumber == reverseNumber)
{
 System.out.println(originalNumber +
 " is a palindrome number");
}
else
{
 System.out.println(originalNumber +
 " is not a palindrome number");
}

```

11] Write a program to perform search operation in an array

→ import java.util.Scanner;

```

class ArraySearch
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 int [] array = {1, 2, 3, 4, 5, 6, 7, 8, 9, 103};
 System.out.println ("Enter the number
 to search :"),
 int target = sc.nextInt();
 }
}
```

```

int index = linearSearch (array, target);
if (index != -1)
{
 System.out.println ("Element " + target +
 " found at index " + index);
}
else
{
 System.out.println ("Element " + target +
 " not found in array");
}
sc.close ();
public static int linearSearch (int []
 array, int target)
{
 for (int i=0, i < array.length; i++)
 {
 if (array [i] == target)
 {
 return i;
 }
 }
 return -1;
}

```



### Assignment No 43

1. What is JVM? JDK & JRE explain in detail?
- 1. JVM :- Java Virtual Machine is a abstract computing machine that enables a computer to run java programs.
- It provides runtime environment in which java byte code can be executed.
  - When you compile a java program it get translate into byte code which is platform independent binary format understand by the JVM.
2. JRE :- Java Runtime Environment is a package of software that provides every thing necessary for running java application on a system except for the development tools.
- It includes the JVM class libraries & other let includes JVM supporting files required to run java application.



- Users who only want to run java application needed to install the JRE on their system.
  - 3) JDK :- Java Development Kit a package of software tools & libraries used by developers to create java application.
  - The JDK also includes additional libraries & API's for development purpose & API's for enabling create, compile, debug & package java appn.
- 2) Explain how scanner class used to input data with example.  
→ The scanner class in java is used to read input from various sources data such as keyboard files or strings.
- ] creating scanner object :-  
To use the scanner class you first need to create a scanner object & specify the input sources.

The most common input sources is

System.in  
import java.util.Scanner;  
Scanner sc = new Scanner(System.in);

2) Reading input :-  
once you have created a scanner object you can use the various methods to read input.

System.out.println("Enter your name");  
String name = sc.nextInt();

3) Handling input errors :-  
It's import to handles input errors gracefully especially when reading input of different types.

eg.

```
System.out.print("Enter a number: ");
if (Scanner.hasNextInt()) {
 int number = sc.nextInt();
 System.out.println("You entered " + number);
}
else
{
 System.out.println("Invalid input
please enter an integer.");
}
```



3

q1] closing the scanner :-  
sc.close();

3] Explain relational & logical operators  
in java with example.

→ Relational Operators :-

class Relational

```
{ public static void main (String [] args)
 {
 int a=5;
 int b=10;
 System.out.println (a==b);
 System.out.println (a!=b);
 System.out.println (a>b);
 System.out.println (a<b);
 System.out.println (a>=b);
 System.out.println (a<=b);
 }
}
```

3

Logical operator :-

class Logical

{

4] Public static void main (String [] args)

```
{ boolean x=true;
 boolean y=false;
 System.out.println (x&y);
 System.out.println (x||y);
 System.out.println (!x);
}
```

Y

q4] Write a program to convert the temp  
from degree to fahrenheit ( $C = \frac{5}{9}(F - 32)$ )

→ import java.util.Scanner;

public static void main (String args[])

```
{ Scanner sc = new Scanner (System.in);
 System.out.println ("Enter Temperature
 in Fahrenheit : ");
 double celsius = (5.0/9.0)
 * (fahrenheit - 32);
 System.out.println ("Temperature in
 Celsius : " + celsius + "C");
 sc.close ();
}
```

3

B



Dr. D. Y. PATIL  
EDUCATION FOUNDATION

5] Write a program to calculate the Lcm of two number.

```

→ import java.util.Scanner;
public class LCMcalculator

 public static void main (String []args)
 Scanner sc = new Scanner (System.in);
 System.out.println ("Enter first number:");
 int num1 = sc.nextInt ();
 System.out.print ("Enter second number:");
 int num2 = sc.nextInt ();
 int lcm = calculateLCM (num1, num2);
 System.out.println ("Lcm of " + num1 + " and " + num2 + " is: " + lcm);
 sc.close ();
}

public static int calculateGCD (int a, int b)
{
 if (b == 0)
 return a;
 else
 return calculateGCD (b, a % b);
}

```

6] Write a program in java to input the value of  $x$  &  $n$  & print the sum of the following series :  $s = 1 + (x+2) \cdot 2! + (2x+3) \cdot 3! + (3x+4) \cdot 4! + \dots$  to  $n$  terms.

```

→ import java.util.Scanner;
public class SeriesSum

 public static void main (String []args)
 {
 Scanner out.print ("Enter the value of x:");
 double x = scanner.nextDouble ();
 System.out.print ("Enter the number of terms (n):");
 int n = scanner.nextInt ();
 double sum = 1.0;
 double term;
 for (int i=1, i<=n, i++)
 term = (i * x + i) / factorial (i);
 }

```

```

sum += term;
}
System.out.println ("sum of the series."
+ tsum);
}

private static double factorial (int num)
{
 if (num == 0 || num == 1)
 {
 return 1;
 }
 else
 {
 return num * factorial (num - 1);
 }
}

```

- 7] Explain the labelled break & labelled continue with example.
- In some programming language like javascript & java, you can use labelled break & labelled continue statements to control the flow of nested loops.
- A labelled break allows you to exit a

specific loop, while a labelled continue allows you to skip rest of the current iteration & move to the next iteration of a particular loop

eg.

```

public class LabeledBreakContinueBreak
{
 public static void main (String [] args)
 {
 outerloop : for (int i=0; i<3; i++)
 {
 innerloop : for (int j=0; j<3; j++)
 {
 System.out.println (i + " " + j);
 if (i == 1 && j == 1)
 {
 break outerloop;
 }
 if (i == 0 && j == 2)
 {
 continue innerloop;
 }
 }
 }
 }
}

```

Q.8] What is jagged array in Java, explain with the help of Example.

A jagged array in Java is an array of arrays where each element of the main array can be an array of different lengths. In other words it is an array of array with irregular size unlike a two dimensional array, where all rows have the same length, jagged arrays allows for varying lengths of rows.

```
public class JaggedArrayExample
{
 public static void main(String[] args)
 {
 int[][] jaggedArray = new int[3][];
 jaggedArray[0] = new int[] {1, 2, 3};
 jaggedArray[1] = new int[] {4, 5, 6, 7};
 jaggedArray[2] = new int[] {8, 9};
 for (int i=0; i < jaggedArray.length; i++)
 {
 for (int j=0; j < jaggedArray[i].length; j++)
 System.out.println(jaggedArray[i][j] + " ");
 }
 }
}
```

System.out.println();  
It moves to the next line for the  
next row



## Assignment No. 4

1. → Explain Execution flow of java program  
The execution flow of a Java program typically follows these steps :
1. compilation :- Java source code is compiled into bytecode using the Java compiler.
  2. Loading :- The Java virtual machine loads the bytecode into memory.
  3. Verification :- The bytecode verifier checks the bytecode to ensure it doesn't violate security constraints.
  4. Preparation :- Memory is allocated for class variables & initialized to default values.
  5. Resolution :- symbols in the bytecode are resolved to references in the runtime constant pool.
  - 6] Initialization :- static variables are initialized, static initialization blocks

are executed, & static methods are invoked.

7. Execution :- The JVM starts executing the main() method of the class specified during startup. The execution flow follows the statements in the main method & any other methods called from it.
8. Termination :- The program terminates when the main method completes execution or when a termination condition is met.

2. Explain how JOptionPane.showInputDialog() used to input data with example.  
→ In Java, JOptionPane.showInputDialog() is used to display a dialog box that prompts the user to enter some data. It returns the data entered by the user as a string.

9. import javax.swing.JOptionPane;  
public class InputExample  
{  
 public static void main (String [] args)  
 {

String userInput = JOptionPane.showInputDialog ("Enter your name.");  
if (userInput == null)  
{  
 System.out.println ("User canceled input.");  
}  
else  
{  
 System.out.println ("You entered :" + userInput);  
}

3) Explain Bitwise operators in java with example.

→ In Java, bitwise operators manipulate individual bits of integer operands.

- There are six bitwise operators :-

1. AND (&) :- performs a bitwise AND operation. It sets each bit to 1 if both corresponding bits of the operands are 1, otherwise sets the bit to 0.

2. OR (|) :- performs a bitwise OR operation.

It sets each bit to 1 if at least one corresponding bit of the operands is 1.



3] **XOR (^)** :- performs a bitwise XOR operation.  
It sets each bit to 1 if exactly one corresponding bit of the corresponds is 1.

4] **NOT (~)** :- performs a bitwise NOT operation. It inverts the bits of its operand. It sets each bit to its complement (0 becomes 1 & 1 becomes 0).

5] **Left shift (<<)** :- shift the bits of the first operand to the left by the no. of positions specified by the second operand.

6] **Right shift (>>)** :- shift the bits of the first operand to the right by the number of positions specified by the second part.

Example :-

```
public class BitwiseExample
```

public static void main (String [] args)

```
int a = 5;
int b = 3;
int resultAnd = a & b;
System.out.println ("AND :" + resultAnd);
int resultOr = a | b;
System.out.println ("OR :" + resultOr);
int resultXor = a ^ b;
System.out.println ("XOR :" + resultXor);
int resultNotA = ~a;
System.out.println ("NOT a :" + resultNotA);
int resultLeftshift = a << 1;
System.out.println ("left shift a by 1 :"
+ resultLeftshift);
int resultRightshift = a >> 1;
System.out.println ("right shift a by 1 :"
+ resultRightshift);
y
y
```

4] Enter any character using Keyboard & find whether it is uppercase, lowercase digit or special symbol.

→ import java.util.Scanner;  
public class CharacterType

```

public static void main (String [] args)
{
 Scanner scanner = new Scanner (System.in);
 System.out.print ("Enter a character :");
 char ch = scanner.next().charAt (0),
 if (Character.isUpperCase (ch))
 System.out.println ("Uppercase letter");
 else if (Character.isLowerCase (ch))
 System.out.println ("Lowercase letter");
 else if (Character.isDigit (ch))
 System.out.println ("Digit");
 else
 System.out.println ("Special symbol");
 scanner.close ();
}

```

5] Enter any three digit number & reverse it using / & % operators.

```

import java.util.Scanner;
public class ReverseNumber
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter a three-digit no.");
 int number = scanner.nextInt();
 if (number >= 100 && number <= 999)
 {
 int digit1 = number / 100;
 int digit2 = (number % 100) / 10;
 int digit3 = number % 10;
 int reverseNumber = (digit1 * 100) +
 (digit2 * 10) + digit3;
 System.out.println ("Reverse number : " +
 reverseNumber);
 }
 else
 System.out.println ("Please enter a
valid three-digit number");
 scanner.close ();
 }
}

```



6) Write a program to check entered number is Disarium Number or not

```
→ import java.util.Scanner;
public class DisariumNumber {
 public static void main (String [] args) {
 Scanner scanner = new Scanner (System.in);
 System.out.print ("Enter a number.");
 int number = scanner.nextInt ();
 if (isDisarium (number))
 System.out.println (number + " is a Disarium number.");
 else
 System.out.println (number + " is not a Disarium number.");
 scanner.close ();
 }
 public static boolean isDisarium (int num) {
 String strNum = String.valueOf (num);
 int length = strNum.length ();
 int sum = 0;
```



for (int i=0; i<length; i++)
 {
 int digit = Character.getNumericValue
 (strNum.charAt (i));
 sum += Math.pow (digit, i+1);
 }
 return sum == num;
}

7) Write a program to find frequency of each digit in a entered integer.

```
→ import java.util.Scanner;
public class DigitFrequency {
 public static void main (String [] args) {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter an integer");
 int number = scanner.nextInt ();
 int [] digitFrequency = new int [10];
 while (number != 0)
 {
 int digit = number % 10;
 digitFrequency [digit]++;
 number /= 10;
 }
 }
}
```

```

System.out.println ("Frequency of each
digit ");
for (int i = 0, is 10, i++)
{
 if (digitFrequency [i] > 0)
 System.out.println ("Digit " + i + ":" +
 digitFrequency [i] + " times");
}

```

Write a prog to check entered number  
is condense Number or not

```
import java.util.Scanner;
class condenseNumber
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter a no:");
 int number = sc.nextInt ();
 if (isCondenseNumber (number))
 System.out.println ("The entered number
correspondence to the pattern of
condense number");
 }
}
```

else  
System.out.println("The entered number  
does not correspond pattern a  
condense number");

```
public static boolean isCondenseNumber
(int Number)
{
```

```
int rowcount = 0;
int totalstar = 0;
```

```

while (totalstars < number)
{
 rowcount++;
 totalstars += rowcount;
}
return totalstars = number;
}

```

ii] E  
 E D  
 E D C  
 E D C B.  
 E D C B A

→ class LetterPattern  
 {  
 public static void main (String [] args)  
 {  
 int n=5;  
 for (int i=0;i<n;i++)  
 {  
 for (int j=n;j>i;j--) \*  
 {  
 System.out.print ((char) ('E'-i)+" ");  
 }  
 System.out.println ();
 }
 }

9. Write a program in Java to find the sum of the given series:  
 $S = a + \frac{1}{2}a^2 + \frac{1}{3}a^3 + \dots + \frac{1}{n}a^n$

→ import java.util.Scanner;  
 public class KhoaiSeries

```

public static void main (String [] args)
{
 Scanner in = new Scanner (System.in);
 System.out.print ("Enter a ");
 int a = in.nextInt ();
 double sum = 0.0;
 for (int i=1;i<=n;i++)
 {
 sum += Math.pow (a,i) / i;
 }
 System.out.println ("Sum = " + sum);
}

```

10. Explain with the help of example, how switch is used within the infinite loop  
 → Using a 'switch' statement within

An infinite loop can be useful for continuously processing input or events & responding accordingly.

Example :-

```
import java.util.Scanner;
public class Main
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 while (true)
 {
 System.out.println ("Entered a command:");
 String input = scanner.nextLine();
 switch (input.toLowerCase ())
 {
 case "start":
 System.out.println ("Starting the process");
 break;
 case "stop":
 System.out.println ("Stopping the process");
 break;
 case "exit":
 System.out.println ("Exiting the program");
 System.exit (0);
 }
 }
 }
}
```

break;  
default:  
System.out.println ("Unknown command.  
Please try again.");  
}  
}

II. Write a prog to compare two arrays & display both having same elements or not.

```
→ import java.util.Arrays;
public class Main
{
 public static void main (String [] args)
 {
 int [] array1 = {1, 2, 3, 4, 5};
 int [] array2 = {5, 4, 3, 2, 1};
 if (arrayHaveSameElements (array1, array2))
 {
 System.out.println ("Both arrays have the
same elements");
 }
 else
 {
 }
 }
}
```

System.out.println("Both arrays do not have the same elements").

```
public static boolean arraysHaveSameElements (int [] array1, int [] array2)
{
 if (array1.length != array2.length)
 return false;
 Arrays.sort (array1);
 Arrays.sort (array2);
 for (int i = 0; i < array1.length; i++)
 if (array1[i] != array2[i])
 return false;
 return true;
}
```

#### Assignment No 5

→ Explain the Java Welcome Program

public class WelcomeProgram

```
public static void main (String [] args)
{
 System.out.println ("Welcome");
}
```

In this program:

- We define a class named 'WelcomeProgram'.
- Inside the class, we have a main method which serves as the entry point of the program.
- Within the 'main' method, we use 'System.out.println()' to print the welcome message "Welcome to Java Programming!" to the console.
- The println method prints the msg & adds a newline character, so the cursor moves to the next line after printing the msg.

2] What is data type, explain different data types in java & using any one way to input data

→ A data type in Java specifies the type of data that a variable can hold.

- Java has two main categories of data types : primitives data types & reference data types.

1] Primitive Data Types : These are the basic data types built into the Java language. They are not objects & have no methods.

There are eight primitive data types in java :

- a) byte :- 8-bit signed integer. It can store values from -128 to 127.
- b) short :- 16-bit signed integer. It can store values from -32,768 to 32767
- c) int :- 32-bit signed integer. It can store values from  $-2^{31} \text{ to } 2^{31}$

- $(2^{31} - 1)$
- d) long :- 64-bit signed integer
  - e) float :- 32-bit floating point integer
  - f) double :- 64-bit floating-point number
  - g) char :- 16-bit Unicode character
  - h) boolean :- Represents true or false values.
- 2] Reference Data Types :-
- These are data types that refer to objects. They are created using predefined classes or user-defined classes.
  - a) String :- Represents a sequence of characters.
  - b) Arrays :- Used to store multiple values of the same type.

Program :-

```

import java.util.Scanner;
public class InputExample
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.println ("Enter an integer:");
 int num = scanner.nextInt ();
 System.out.println ("You entered :" + num);
 scanner.close ();
 }
}

```

3] Explain conditional operator in java with example.

→ In java conditional of also known as ternary operator.

Syntax :- condition ? expression 1:  
                          expression 2

e.g class condition opeg

```

public static void main (String [] args)
{
 int number = 10;
}

```

```

String result = (number % 2 == 0) ? "Even" : "Odd";
System.out.println ("The number is " + result);

```

4] Write a c program to check whether number is -ve, +ve or zero.

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
int num;
```

```
printf ("Enter a no:");
```

```
scanf ("%d", &num);
```

```
if (num < 0)
```

```
{
```

```
 printf ("The number is -ve \n");
```

```
else if (num > 0)
```

```
{
```

```
 printf ("The number is +ve \n");
```

```
}
```

```
else
```

```
{
```

```
 printf ("The number is zero \n");
```

```
}
```

return 0;

5] Entered the marks obtained in 5 subjects calculate the %

```

→ import java.util.Scanner;
class calculatepercentage
{
 public static void main (String [] args)
 {
 Scanner sc= new Scanner (System.in);
 System.out.println("Enter marks obtained
 in 5 sub!");
 s.o.p ("Subject 1");
 int subject1 = sc.nextInt();
 s.o.p ("Sub 2");
 int subject2 = sc.nextInt();
 s.o.p ("Sub 3");
 int subject3 = sc.nextInt();
 s.o.p ("Sub 4");
 int subject4 = sc.nextInt();
 s.o.p ("Sub 5");
 int subject5 = sc.nextInt();
 int total_marks = subject1 + subject2 +
 subject3 + subject4 + subject5;
 }
}

```

double percentage = (double) total\_marks / 5;
 s.o.p ("total marks: " + total\_marks);
 s.o.p ("Percentage: " + percentage + "%");
 sc.close();

6] Write a program to check entered number is happy number or not

```

→ import java.util.Scanner;
class HappyNumber

```

```

public static void main (String args [])
{
 public static boolean isHappy (int n)
 {
 HashSet < Integer > set = new HashSet < Integer >;
 while (n != 1 && !set.contains (n))
 {
 set.add (n);
 n = getnextnumber (n);
 }
 return n == 1;
 }
 private static int getnextnumber (int n)
 {
 int sum = 0;
 while (n > 0)
 {
 ...
 }
 }
}

```

```

 {
 int digit = n % 10;
 sum += digit * digit;
 n /= 10;
 }
 return sum;
}

public static void main (String [] args)
{
 int num = 19;
 if (isHappy (num))
 {
 System.out.println ("num" + " is a happy number");
 }
 else
 {
 System.out.println ("num" + " is not happy number");
 }
}

```

- 7] Write a program to find the  $x^y$  using  
 do while () loop.  
 → import java.util.Scanner;

```

class main
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter the base(x)");
 int x = scanner.nextInt ();
 System.out.print ("Enter the exponent (y)");
 int y = scanner.nextInt ();
 int result = 1;
 int count = 0;
 while (count < y)
 {
 result *= x;
 count++;
 }
 System.out.println (x + " raised to the power of " + y + " is " + result);
 }
}

```

- 8] Write a program to check entered number is condehse number or not.



```

import java.util.Scanner;
class condenseNumber
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter a number : ");
 int number = sc.nextInt ();
 if (isCondenseNumber (number))
 System.out.println (number + " is a condense no");
 else
 System.out.println (number + " is not a condense no");
 }
}

```

Dr. D. Y. PATH.  
INTERCOLLEGIATE INSTITUTE

```

public static boolean isCondenseNumber
(int num)
{
 int square = num * num;
 String squareStr = Integer.toString (square);
 String numStr = Integer.toString (num);
 int i = 0, j = 0;
 while (i < numStr.length () & j < squareStr.length ())
 {
 if (numStr.charAt (i) == squareStr.charAt (j))
 j++;
 i++;
 }
 return i == numStr.length ();
}

```

if (numStr.charAt (i) == squareStr.charAt (j))  
 {  
 j++;  
 }  
 i++;  
} else  
 {  
 return i == numStr.length ();
 }

ii]  
 A  
 B C  
 D E F  
 G H I J  
 K L M N O



```

→ import java.util.Scanner;
class concentricnumbercheck {
 public static void main (String [] args) {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter the number to check:");
 int number = sc.nextInt();
 if (isconcentricnumber (number)) {
 System.out.println (number + " is a
 concentric number.");
 } else {
 System.out.println (number + " is not a concentric
 number.");
 }
 }

 public static boolean isconcentricno
 (int number) {
 int squareRoot = (int) Math.sqrt
 (number);
 int totalElements = squareRoot * squareRoot;
 int midElement = totalElement / 2 + 1;
 return number == midElement;
 }
}

```

9]

Write a program in to display the first ten terms of the following series:

→ 1, 4, 9, 16  
 class mainc {
 public static void main (String [] args) {
 for (int i=1; i<=10; i++) {
 int term = i\*i;
 System.out.print ("term + " );
 }
 }
 }

10]

Write a prog using the switch case to demonstrate how use a character case constant.

→ import java.util.Scanner;
class charactercasdemo {
 public static void main (String [] args) {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter a character:");
 char ch = sc.next () - charAt (0);
 switch (character to uppercase (ch))
 }
}

```

}
case 'A':
 System.out.println("The character is an uppercase A");
 break;
case 'B':
 System.out.println("The character is an uppercase B");
 break;
case 'C':
 System.out.println("The character is an uppercase C");
 break;
default:
 System.out.println("The character is not A,B or C");
}
System.out.close();

```

y

ii] Write a program to find repetition of entered number in an array.

→ import java.util.Scanner;

```

class main
{
 public static void main (String [] args)
 {
 Scanner sc = new Scanner (System.in);

```

```

System.out.println ("Enter the size of
 the array ");
int size = sc.nextInt ();
int [] arr = new int [size];
System.out.println ("Enter the elements
 of the array ");
for (int i=0; i<size; i++)
{
 arr [i] = sc.nextInt ();
}

System.out.println ("Enter the number
 to find repetition ");
int target = sc.nextInt ();
int count = 0;
for (int num : arr)
{
 if (num == target)
 {
 count++;
 }
}
System.out.println ("The number " + target +
 " repeats " + count + " times in the
 array ");

```



Dr. D. Y. PATIL  
EDUCATIONAL FEDERATION

so close ( )

y

y