Academic Year: 2023-24 **Semester: II** **Class: FYMCA**

Course Code: MC506 **Course Name: Java Programming**

Name: Durgesh Dilip Mandge **UCID : 2023510032**

Experiment No.1.1

Date: 05-02-2024

Aim: Fundamentals of Java Programming

CO Mapping – CO 1

Objective:

- To understand declaration of Classes, and Methods with its all features such as Constructors, Access Specifier
- To understand Classes, Instance variables, Methods, Constructors, Access
- Specifiers as basic fundamentals
- Implement Abstract Classes and Wrapper Classes for given problem statement
- Design and implement Inheritance, Polymorphism in JAVA
- Demonstrate Use of Static, final, super and this keyword
- Demonstrate creating user defined package, Access control protection,
- Defining interface, Implementing interface

Lab Exercise: Solve the various problem statement using conditional statements, loops and some input functions.

Code:

```
package Java;

import java.util.Scanner;

public class Lab1{
    public static void main(String[] args) {
        // multiplicationTable(5);
        // System.out.println(gcd(80, 120));
        // System.out.println(calculator(34, 12, '/'));
        // fibo(0);
        // reverse("Durgesh");
        // Scanner s = new Scanner(System.in);
        // System.out.println(s.nextLine());
        // s.close();;
        // triangle(5);
        // System.out.println(octalToDecimal(132));
        // prime(10);
        // printDigits(15089);
        // PersonalDeta();
        // System.out.println(palindrome(1312));.
        // age(04,06,2002);
        divisionOfNo();
    }

    //1.      To Generate Multiplication Table
    static void multiplicationTable(int a){
        int c =10;
        while(c-->0){
            System.out.print(a*(10-c) + " ");
        }
    }

    //2.      To Find GCD of two Numbers
    static int gcd(int a, int b){
        int gcd =1;
        for(int i =1; i<=a && i<=b ; i++){
            if(a%i==0 && b%i==0){
                gcd=i;
```

```
            }
        }
        return gcd;
    }

    //3.      Calculator Program in Java
    static int calculator(int a, int b, char ch){
        switch (ch) {
            case '+':
                return a+b;
            case '-':
                return a-b;
            case '*':
                return a*b;
            case '/':
                return a/b;

            default:
                return 0;
        }
    }

    //4.      To calculate Fibonacci Series up to n numbers.
    static void fibo(int a){
        int first =0, second=1;
        if(a==0){
            System.out.print("0 ");
            return;
        }
        if(a==1){
            System.out.print("0 1");
            return;
        }
        System.out.print("0 ");
        for(int i=1; i<a; i++){
            System.out.print(second + " ");
            int t = first;
            first=second;
            second+=t;
        }
```

```
    }

    //5.      W.A.P to reverse your First Name using Strings .
    static void reverse(String name){
        char[] c = name.toCharArray();
        for(int i=0; i<c.length/2; i++){
            swap(i,c.length-1-i,c);
        }
        System.out.println(c.toString());
    }


    private static void swap(int f, int s, char[] c) {
        char temp = c[f];
        c[f]=c[s];
        c[s]=temp;
    }

    //6.      W.A.P to Print one number at a time , input must be from the
user
    static void printDigits(int i){
        int[] arr=new int[5];int j=0;
        while(i>0){
            arr[j]=i%10;
            j++;
            i/=10;
        }
        for(int k=j-1;k>=0;k--){
            System.out.println(arr[k]);
        }
    }

    //7.      W.A.P to print your Personal details ( Name , Gender,
Address , Phone No.,College Name )
    static void PersonalDeta(){
        Scanner s = new Scanner(System.in);
        System.out.println("Name: ");
        String name = s.next();
        System.out.println("Gender: ");
        String gender = s.next();
```

```
        System.out.println("Address ");
        String add = s.next();
        System.out.println("Phone No ");
        String phone = s.next();
        System.out.println("College Name ");
        String colg = s.next();
        System.out.println("Your name is " + name);
        System.out.println("Your geder is " + gender);
        System.out.println("Your address is " + add);
        System.out.println("Your phone is " + phone);
        System.out.println("Your college is " + colg);
    }

    //8.      W.A.P to check whether a number is  Odd or even
    static boolean isOdd(int n){
        return n%2==0;
    }

    //9.       W.A.P to check whether a number is palindrome or Not.
    static boolean palindrome(int n){
        int reverse = 0;
        int original = n;
        while(n>0){
            reverse*=10;
            reverse+=(n%10);
            n/=10;
        }
        return reverse==original;
    }

    //10.     W.A.P to add 10  numbers of one series .(Example :- If the
user inputs 3 then its should take  numbers from 3,4,5……..12)
    static void series(int n){
        for(int i=0;i<10;i++){
            System.out.println(n+i);
        }
    }

    //11.      W.A.P to print your Age based on your Birth date
    static void age(int date, int month, int year){
```

```
        int cdate=23, cmonth=1, cyear=2024;
        System.out.println("You are "+ (cdate-date) + "days " +
(month-cmonth) + "months " + (cyear-year-1) + "years old.");
    }

    // 12.      W.A.P to create  the following output :-
    //          1
    //          11
    //          111
    //          1111
    //          11111
    static void triangle(int n){
        for(int i=0; i<n; i++){
            for(int j=0;j<=i; j++){
                System.out.print(1);
            }
            System.out.println();
        }
    }

    //13.  W.A.P to accept any two numbers and perform division on it ( If
the number is in decimal value then convert them into a whole number)
    static void divisionOfNo(){
        Scanner sc = new Scanner(System.in);
        float f = sc.nextFloat();
        float s = sc.nextFloat();
        System.out.println((int)(f/s));
    }

    //14.  W.A.P to convert number in characters (E.g. 123 , Output One
Two Three )
    private static final String[] digitWords = {
        "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven",
"Eight", "Nine"
    };

    public static void convertNumberToCharacters(int number) {
        String numStr = String.valueOf(number);
        for (int i = 0; i < numStr.length(); i++) {
            int digit = Character.getNumericValue(numStr.charAt(i));
```

```
            System.out.print(digitWords[digit] + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        System.out.print("Equivalent characters: ");
        convertNumberToCharacters(number);
    }

    // 15.  To convert Number to word
    private static final String[] units = {"", "One", "Two", "Three",
"Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten",
            "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen",
"Sixteen", "Seventeen", "Eighteen", "Nineteen"};

    private static final String[] tens = {"", "", "Twenty", "Thirty",
"Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

    public static String convertToWord(int number) {
        if (number == 0) {
            return "Zero";
        }
        return convert(number);
    }

    private static String convert(int number) {
        if (number < 20) {
            return units[number];
        }
        if (number < 100) {
            return tens[number / 10] + ((number % 10 != 0) ? " " : "") +
units[number % 10];
        }
        if (number < 1000) {
            return units[number / 100] + " Hundred" + ((number % 100 != 0)
? " and " : "") + convert(number % 100);
```

```
        }
        if (number < 1000000) {
            return convert(number / 1000) + " Thousand" + ((number % 1000
!= 0) ? " " : "") + convert(number % 1000);
        }

    // 16.  Java Program to Check Whether a Number is Prime or Not using
different control structurepublic static boolean isPrime(int num) {
        public static boolean isPrime(int num) {
            if (num <= 1) {
                return false;
            }
            for (int i = 2; i <= Math.sqrt(num); i++) {
                if (num % i == 0) {
                    return false;
                }
            }
            return true;
        }

    // 17. To Check a Leap year
    static boolean leap(int n){
        return n%4==0;
    }

    // 18. To check whether a number is positive or negative
    static boolean isPositive(int n){
        return n>0;
    }

    // 19. To calculate the sum of Natural Numbers
    static int sum(int i){
        return i*(i+1)/2;
    }

    // 20. To Find the factorial of a Number
    static int fac(int i){
        if(i<=1){
            return i;
        }
```

```
        return i*fac(i-1);
    }

    // 21.To display all prime numbers from 1 to N.
    static void prime(int n){
        for(int i=1; i<=n; i++){
            if(isPrime(i)){
                System.out.println(i);
            }
        }
    }

    private static boolean isPrime(int n) {
        int i =2;
        while(i*i<n){
            if(n%i==0){return false;}
            i++;
        }
        return true;
    }

    // 22.To check whether Input character is Vowel or Not.
    static boolean isVowel(char ch){
        return ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u';
    }

    // 23.To calculate simple Interest
    static int simpleInterest(int amount,int rate,int years){
        return amount*rate*years/100;
    }

    // 24.To convert octal to decimal conversion
    static int octalToDecimal(int oct){
        int base=1, ans=0;
        while(oct>0){
            ans+=(oct%10)*base;
            base*=8;
            oct/=10;
        }
        return ans;
```

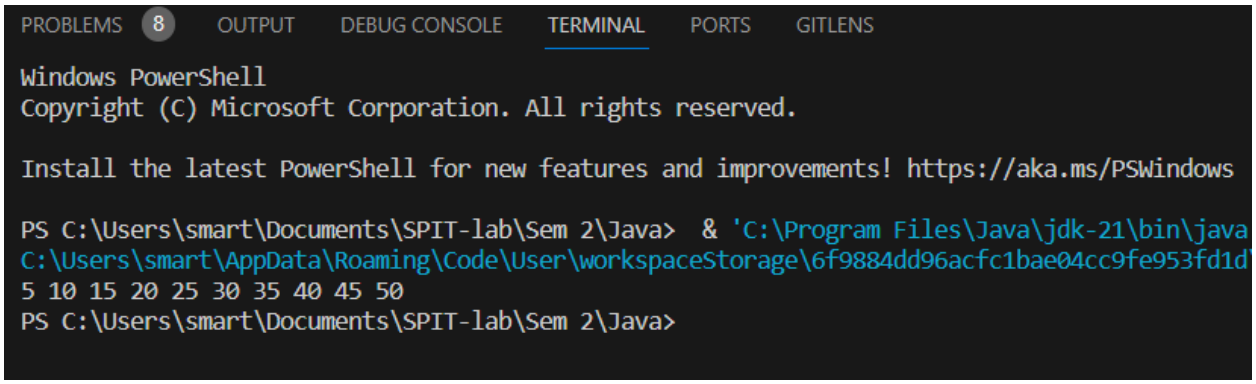
```
    }

    // 25. Conclude what is the difference between JAVA and C++.
    //1. Java do not have concept of pointers it is all call by reference
by default
    //2. CPP code is compiled and converted directly into native machine
code while
    //    java code is compiled to form intermidiate bytecode which is
portable over various platforms
    //3. Java support in built library for various tasks like networking,
file I/O, collections, GUI development
    //    CPP also has STL but it's not as comprehensive as Java's
standard library.
    //4. Memory management is abstract in java while in CPP it is manual
    // etc.

}
```

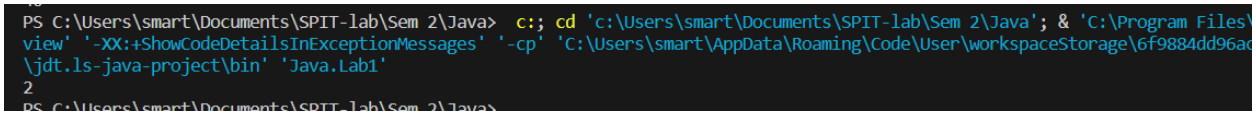
Outputs:

1. **Multiplication Table**



2. **gcd(80,120)**

3. **calculator(24,12,'/')**



4. **fibo(7)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart\Do
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart\AppDa
\jdt.ls-java-project\bin' 'Java.Lab1'
0 1 1 2 3 5 8
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

5. **reverse("Durgesh")**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart\Documents\SPIT-la
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart\AppData\Roaming\Code
\jdt.ls-java-project\bin' 'Java.Lab1'
hsegruD
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

6. **printDigits(15089)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart\
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart\App
\jdt.ls-java-project\bin' 'Java.Lab1'
1
5
0
8
9
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

7. **personalData()**

```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS

Name:
DurgeshMandge
Gender:
Male
Address
Mumbai
Phone No
1122334455
College Name
SPIT
Your name is DurgeshMandge
Your geder is Male
Your address is Mumbai
Your phone is 1122334455
Your college is SPIT
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

8. **isOdd(34)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\s
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smar
\jdt.ls-java-project\bin' 'Java.Lab1'
true
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

9. **isPalindrome(1312)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\sm
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart
\jdt.ls-java-project\bin' 'Java.Lab1'
false
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

10. **Series(10)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:;
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
\jdt.ls-java-project\bin' 'Java.Lab1'
10
11
12
13
14
15
16
17
18
19
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

11. **age(04,06,2002)**

```
Install the latest PowerShell for new features and improvements! https://aka.ms/

PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  & 'C:\Program Files\Java\jdk-
C:\Users\smart\AppData\Roaming\Code\User\workspaceStorage\6f9884dd96acfc1bae04c
You are 19days 5months 21years old.
```

12. **triangle(5)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart\Documents
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart\AppData\Roam
\jdt.ls-java-project\bin' 'Java.Lab1'
1
11
111
1111
11111
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

13. **Division of two numbers**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  & 'C:
nExceptionMessages' '-cp' 'C:\Users\smart\AppData\Roami
jdt.ls-java-project\bin' 'Java.Lab1'
24 12
2
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

14. **Number to string characters**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart\Documents\SPIT-lab\s
bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\sm
ge\6f9884dd96acfc1bae04cc9fe953fd1d\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'Java.Lab1'
Enter a number: 34
Equivalent characters: Three Four
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

15. **NumberToWord**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart\Documer
bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp
ge\6f9884dd96acfc1bae04cc9fe953fd1d\redhat.java\jdt_ws\jdt.ls-java-project\bin'
Thirty Four
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

16. **isPrime(34)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\sm
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart
\jdt.ls-java-project\bin' 'Java.Lab1'
false
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

17. **isLeap(2023)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\sm
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart
\jdt.ls-java-project\bin' 'Java.Lab1'
false
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

18. **isPositive(-12)**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\sm
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart
\jdt.ls-java-project\bin' 'Java.Lab1'
false
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

19. **sum(34)**

```
PROBLEMS 6  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smar
bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessag
ge\6f9884dd96acfc1bae04cc9fe953fd1d\redhat.java\jdt_ws\jdt.ls-java-proj
595
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

20. **factorial(4)**

```
PROBLEMS 6  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart
bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessage
ge\6f9884dd96acfc1bae04cc9fe953fd1d\redhat.java\jdt_ws\jdt.ls-java-proje
24
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

21. **prime(15)**

```
PS C:\Users\smart\Documents\SPIT-lab\
bin\java.exe' '--enable-preview' '-XX
ge\6f9884dd96acfc1bae04cc9fe953fd1d\r
2 3 5 7 11 13
PS C:\Users\smart\Documents\SPIT-lab\
```

22. **isVowel('P')**

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\sm
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart
\jdt.ls-java-project\bin' 'Java.Lab1'
false
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

23. **simpleInterest(1000,3,8)**

```
2 3 5 7 11 13
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\U
bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExcept
ge\6f9884dd96acfc1bae04cc9fe953fd1d\redhat.java\jdt_ws\jdt.ls-
240
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

24. octalToDecimal(132)

```
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>  c:; cd 'c:\Users\smart
view' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smart\Ap
\jdt.ls-java-project\bin' 'Java.Lab1'
90
PS C:\Users\smart\Documents\SPIT-lab\Sem 2\Java>
```

25. Conclude what is the difference between JAVA and C++.

1. Java do not have concept of pointers it is all call by reference by default
2. CPP code is compiled and converted directly into native machine code while java code is compiled to form intermidiate bytecode which is portable over various platforms
3. Java support in built library for various tasks like networking, file I/O, collections, GUI development CPP also has STL but it's not as comprehensive as Java's standard library.
4. Memory management is abstract in java while in CPP it is manual etc.

Observation:

Java is the statically typed and data safe language. We get abstraction of data, encapsulation of logic and polymorphism of features which makes java Object oriented. Every function has some return type and input parameters on which it performs its calculations. We can use conditional statements and loops for achieving complex logic solution in java.