

Course Code 13044		
LAB : Java Programming		
	Contents	Page No
Unit 1 to 4	BLOCK 1 JAVA FUNDAMENTALS	
1	Display "Hello, Welcome to Java Programming"	1
2	Store and access values	2
3	A Simple profile printing from user input	3
4	Factorial for the given number	5
5	Simple Calculator	6
6	Student mark sheet preparation	8
7	Smallest and Biggest number from n numbers	10
8	Multiplication table	11
Unit 5 to 8	BLOCK 2 : OOP CONCEPTS	
9	Function overloading	12
10	Understanding inheritance in java	14
11	Packages, creating own class functions into package	16
Unit 9 - 10	BLOCK 3 : VIRTUAL FUNCTIONS AND THREADS	
12	Java Threads, using two different threads to print odd and even number.	18
13	Single thread class multiple task	20
14	Virtual function	22
Unit 11 to 12	BLOCK 4 : I/O AND EXCEPTION HANDLING	
15	Exception Handling, Monitoring catching errors or events or any changes and redirection the execution.	23
16	User defined exception, finding not a number and throw exception	24
17	I/O in Java (Reading a text file) using scanner class	25
18	I/O in Java - Writing into a text file using FileWriter class	26
19	I/O in Java - Writing into a text file using FileWriter class	27
Unit 13 - 14	BLOCK 5 : I/O APPLET AND NETWORK PROGRAMMING	
20	Java Applet – A simple program using NETBEANS IDE	28
21	Applet – Embedding with html	29
22	Networking – Retrieving the current system IP Address	30
23	Networking – Retrieving the host system IP Address	31
24	Networking – Connection establishment between Client and Server	32
25	Networking – Chat text between Client and Server	34

1. Display “Hello, Welcome to Java Programming”

Objective : Understanding how to display text on the screen

Program :

```
public class JavaLabPrograms {  
  
    public static void main(String[] args) {  
        // Display message on the screen  
        System.out.println("Hello, Welcome to Java Programming.");  
    }  
  
}
```

Output :

Hello, Welcome to Java Programming.

2. Store and access values

Objective : Understanding to store and access values

Program :

```
public class JavaLabPrograms {  
  
    public static void main(String[] args) {  
        // Display my Address, values from variables  
        System.out.println("Display my Address, values from variables");  
        String myAddress = "K.Sekar \nNo : 546 4th Cross Street \nCoimbatore 641046";  
        System.out.println(myAddress);  
    }  
  
}
```

Output:

Display my Address, values from variables

K.Sekar

No : 546 4th Cross Street

Coimbatore 641046

3. Simple profile printing from user input

Objective : Understanding Different datatypes and formatted output

Program :

```
public class JavaLabPrograms {
    public static void main(String[] args) {
        // A Simple profile printing from user input
        String name;
        int age;
        float weight;
        char gender;
        Scanner kinput = new Scanner(System.in);
        System.out.println("Profile Preparation Program");
        System.out.println("Type you details as follows");
        System.out.println("Type your name : ");
        name = kinput.nextLine();
        System.out.println("Type your age : ");
        age = kinput.nextInt();
        System.out.println("Type your weight : ");
        weight = kinput.nextFloat();
        System.out.println("Type your gender : ");
        gender = kinput.next().charAt(0);
        System.out.println(" My Profile ");
        System.out.println(" -----");
        System.out.println("Name  : " + name);
        System.out.println("Age   : " + String.valueOf(age));
        System.out.println("weight : " + String.valueOf(weight));
        System.out.println("gender : " + gender);
    }
}
```

Input :

Profile Preparation Program

Type you details as follows

Type your name :

pals

Type your age :

25

Type your weight :

36.36

Type your gender :

M

Output :

My Profile

Name : pals

Age : 25

weight : 36.36

gender : m

4. Factorial for the given number

Objective : understanding loop construction

Program :

```
public class JavaLabPrograms {  
  
    public static void main(String[] args) {  
        // Find factorial for the given number  
        Scanner kinput = new Scanner(System.in);  
        System.out.println("Find the factorial for the given number");  
        System.out.println("-----");  
        System.out.println("Enter the number :");  
        int number = kinput.nextInt();  
        int result =1;  
        for(int i = 1; i <= number; i++)  
        {  
            result = result * i;  
        }  
  
        System.out.println("Factorial for " + number + " is : " + result);  
    }  
}
```

Input :

Find the factorial for the given number

Enter the number :

7

Output :

Factorial for 7 is : 5040

5. Simple Calculator

Objective : understanding conditional statements and arithmetic expressions

Program :

```
public class JavaProgramsAll {
    public static void main(String[] args) {
        // Simple calculator
        char operator = '+';
        double operand1, operand2, result = 1;
        Scanner kinput = new Scanner(System.in);
        System.out.println("Simple Calculator ");
        System.out.println("-----");
        System.out.println("Enter the operand 1 (value) : ");
        operand1 = kinput.nextDouble();
        System.out.println("Enter operator ");
        operator = kinput.next().charAt(0);
        System.out.println("Enter the operand 1 (value) : ");
        operand2 = kinput.nextDouble();
        switch(operator)
        {
            case '+':
            {
                result = operand1 + operand2;
                break;
            }
            case '-':
            {
                result = operand1 - operand2;
                break;
            }
            case '*':
            {
                result = operand1 * operand2;
                break;
            }
            case '/':
            {
                result = operand1 / operand2;
                break;
            }
            default :
            {
                System.out.print("Please check input values , Example \n56\n+\n24\n\n");
                return;
            }
        }
    }
}
```

```
    }  
  }  
  if(result!=1)  
  {  
    System.out.println(result);  
  }  
}  
  
}
```

Input :

Simple Calculator

Enter the operand 1 (value) :

26

Enter operator

+

Enter the operand 1 (value) :

12

Output:

26.0 + 12.0 = 38.0

6 Student mark sheet preparation

Objective : Input the student name and marks, and print the pass/fail status for individual status, total and average.

Program :

```
public class JavaProgramsAll {

    public static void main(String[] args) {
        // SSLC Student Marksheet
        String name;
        float average;
        int language1, language2, maths, science, socialscience, total;
        Scanner kinput = new Scanner(System.in);
        System.out.println("SSLC Student marksheet calculation");
        System.out.print("\n-----");
        System.out.print("\nEnter the name :");
        name = kinput.nextLine();
        System.out.print("\nEnter the Language 1 mark    : ");
        language1 = kinput.nextInt();
        System.out.print("\nEnter the Language 2 mark    : ");
        language2 = kinput.nextInt();
        System.out.print("\nEnter the Mathematics marks    : ");
        maths = kinput.nextInt();
        System.out.print("\nEnter the Science marks      : ");
        science = kinput.nextInt();
        System.out.print("\nEnter the Social science marks : ");
        socialscience = kinput.nextInt();
        total = language1 + language2 + maths + science +socialscience;
        System.out.println("-----SSLC Student Marksheet----- ");
        System.out.println("-----");
        System.out.println("Name      : " + name);
        System.out.println("Subjects   Marks      Status");
        System.out.println("-----");
        System.out.print("Language 1    " + language1 + "      ");
        if(language1 > 39) System.out.print("Pass"+ "\n"); else System.out.print("Fail" + "\n");
        System.out.print("Language 2    " + language2 + "      ");
        if(language2 > 39) System.out.print("Pass"+ "\n"); else System.out.print("Fail"+ "\n");
        System.out.print("Mathematics   " + maths    + "      ");
        if(maths > 39) System.out.print("Pass"+ "\n"); else System.out.print("Fail"+ "\n");
        System.out.print("Science       " + science   + "      ");
        if(science > 39) System.out.print("Pass"+ "\n"); else System.out.print("Fail"+ "\n");
        System.out.print("Social Science " + science   + "      ");
        if(socialscience > 39) System.out.print("Pass"+ "\n"); else System.out.print("Fail"+ "\n");
        System.out.println("Total    = " + total);
        System.out.println("average = " + total/5);
    }
}
```

```

if(language1 > 39 && language2 > 39 && maths >39 && science>39 && socialscience>39)
    System.out.println("Overall Result : PASS" );
else
    System.out.println("Overall Result : FAIL" );
System.out.println("-----");
}
}

```

Input :

SSLC Student marksheet calculation

```

-----
Enter the name :pals
Enter the Language 1 mark    : 65
Enter the Language 2 mark    : 59
Enter the Mathematics marks  : 89
Enter the Science marks      : 88
Enter the Social science marks : 93

```

Output:

```

-----SSLC Student Marksheet-----
-----
Name      : pals
Subjects  Marks      Status
-----
Language 1  65        Pass
Language 2  59        Pass
Mathematics 89        Pass
Science     88        Pass
Social Science 88      Pass
Total = 394
average = 78
Overall Result : PASS
-----

```

7. Smallest and Biggest number from n numbers

Objective :

Using array to store and access, conditional statement to find the biggest and smallest number. Set the array size (n numbers input) from user.

Program :

```
import java.util.Scanner;
public static void main(String[] args) {
    // Find biggest and smallest number from the given numbers
    Scanner kinput = new Scanner(System.in);
    System.out.println("Enter the total input numbers");
    int totalnumbers, smallno, bigno;
    totalnumbers = kinput.nextInt();
    int inputnumbers[] = new int[totalnumbers];
    for(int i=0;i <totalnumbers; i++)
    {
        System.out.println("Enter number : ");
        inputnumbers[i] = kinput.nextInt();
    }
    smallno = inputnumbers[0];
    bigno = inputnumbers[0];
    System.out.println(" The numbers you entered : ");
    for(int i=0;i <totalnumbers; i++)
    {
        if(smallno > inputnumbers[i]) smallno = inputnumbers[i];
        if(bigno < inputnumbers[i]) bigno = inputnumbers[i];
        System.out.print(inputnumbers[i] + " ");
    }
    System.out.println(" \nThe Biggest no is : " + bigno);
    System.out.println(" The smallest no is : " + smallno);
}
}
```

Input :

```
Enter the total input numbers : 6
Enter number : 56
Enter number : 65
Enter number : 88
Enter number : 99
Enter number : 77
Enter number : 44
```

Output :

```
The numbers you entered :
56 65 88 99 77 44
The Biggest no is : 99
The smallest no is : 44
```

8. Multiplication table

Objective : understanding the looping construction, arithmetic expression and formatted output

Program :

```
public class JavaProgramsAll {
    public static void main(String[] args) {
        // Printing the multiplication table

        int tableno, limit;
        Scanner kinput = new Scanner(System.in);
        System.out.println("Program to print multiplication table");
        System.out.print("Enter the table no : ");
        tableno = kinput.nextInt();
        System.out.print("\nEnter the iteration limit (upto) :");
        limit = kinput.nextInt();
        System.out.println("\n\n -----");
        System.out.println("Selected Table No : " + tableno + " \n and Iteration upto " + limit);
        for(int i=1;i <= limit; i++)
        {
            System.out.println(i + " X " + tableno + " = " + (i*tableno));
        }
        System.out.println("\n -----");
    }
}
```

Input :

Program to print multiplication table

Enter the table no : 8

Enter the iteration limit (upto) :5

Output:

Selected Table No :8

and Iteration upto 5

1 X 8 = 8

2 X 8 = 16

3 X 8 = 24

4 X 8 = 32

5 X 8 = 40

9. Function overloading

Objective : using the same function name with different signatures

Program :

```
public class JavaProgramsAll {
    public static void main(String[] args) {
        //Method Overloading - Adding two numbers

        System.out.println(" Program for Method Overloading ");
        System.out.println("Adding two numbers using same function name with different (datatypes)
arguments");
        System.out.println(addtion(12,      18));
        System.out.println(addtion(47,      (float) 53.26));
        System.out.println(addtion((float) 93.75,  32));
        System.out.println(addtion((float) 159.84, (float) 123.42));
    }
    public static int addtion(int num1, int num2)
    {
        System.out.println("This is function from \"(addtion(int num1, int num2)\");
        System.out.print("Input is " + num1 + " + " + num2 + " \nOutput is ");
        return num1 + num2;
    }
    public static float addtion(float num1, int num2)
    {
        System.out.println("This is function from \"addtion(float num1, int num2)\");
        System.out.print("Input is " + num1 + " + " + num2 + " \nOutput is ");
        return num1 + num2;
    }
    public static float addtion(int num1, float num2)
    {
        System.out.println("This is function from \"addtion(int num1, float num2)\");
        System.out.print("Input is " + num1 + " + " + num2 + " \nOutput is ");
        return num1 + num2;
    }
    public static float addtion(float num1, float num2)
    {
        System.out.println("This is function from \"addtion(float num1, float num2)\");
        System.out.print("Input is " + num1 + " + " + num2 + " \nOutput is ");
        return num1 + num2;
    }
}
```

Input

12 , 18

47 , 53.26

93.75 , 32

159.84, 123.42

Output

Program for Method Overloading

Adding two numbers using same function name with different (datatypes) arguments

This is function from "(addtion(int num1, int num2))"

Input is 12 + 18

Output is 30

This is function from "addtion(int num1, float num2)"

Input is 47 + 53.26

Output is 100.259995

This is function from "addtion(float num1, int num2)"

Input is 93.75 + 32

Output is 125.75

This is function from "addtion(float num1, float num2)"

Input is 159.84 + 123.42

Output is 283.26

10. Understanding inheritance in java

Objective : understanding class relationship in inheritance

Program :

```
public class JavaProgramsAll {
    public static void main(String[] args) {
        //Inheritance
        HSC hsc = new HSC();
        hsc.input_sslc_details();
        hsc.input_hsc_details();
        hsc.display_sslc_details();
        hsc.display_hsc_details();
    }
}

class SSLC
{
    int sslc_total_marks;
    float sslc_percentage;
    void input_sslc_details()
    {
        Scanner kinput = new Scanner(System.in);
        System.out.println("Enter your SSLC total marks    :");
        sslc_total_marks = kinput.nextInt();
        System.out.println("Enter your SSLC percentage is :");
        sslc_percentage = kinput.nextFloat();
    }
    void display_sslc_details()
    {
        System.out.println("-----");
        System.out.println("Your SSLC total    marks    : " + sslc_total_marks);
        System.out.println("Your SSLC percentage is    : " + sslc_percentage);
        System.out.println("-----");
    }
}

class HSC extends SSLC
{
    int hsc_total_marks;
    float hsc_percentage;
    void input_hsc_details()
    {
        Scanner kinput = new Scanner(System.in);
```

```

        System.out.println("Enter your HSC total marks    : ");
        hsc_total_marks = kinput.nextInt();
        System.out.println("Enter your HSC percentage  is : ");
        hsc_percentage = kinput.nextFloat();
    }
    void display_hsc_details()
    {
        System.out.println("-----");
        System.out.println("Your HSC total    marks    : " + hsc_total_marks);
        System.out.println("Your HSC percentage is    : " + hsc_percentage);
        System.out.println("-----");
    }
}

```

Input

Program for Java class Inheritance
Enter your SSLC total marks : 450
Enter your SSLC percentage is : 90
Enter your HSC total marks : 1127
Enter your HSC percentage is : 187.33

Output :

```

-----
Your SSLC total    marks    : 450
Your SSLC percentage is    : 90.0
-----
-----
Your HSC total    marks    : 1127
Your HSC percentage is    : 187.33
-----

```


11. Packages, creating own class functions into package

Objective : understanding how to pack the related classes into single unit.

Program :

```
import mypackages.*;
public class JavaProgramsAll {
    public static void main(String[] args) {
        //Packages . . .
        MyResidensialAddress myResidensialAddress = new MyResidensialAddress();
        MyOfficeAddress myOfficeAddress = new MyOfficeAddress();
        myResidensialAddress.DisplayAddress();
        myOfficeAddress.DisplayAddress();
    }
}
```

Add class file 1 into this package

```
package mypackages;
```

```
public class MyOfficeAddress {
    String myOfficeAddress = "No 566, Indian Overseas Bank \n Airport Campus,, \nChennai 641538";
    public void DisplayAddress()
    {
        System.out.println("\n-----\n");
        System.out.println("\nFrom mypackages Class 2 :\nMy Office Address is \n" + myOfficeAddress);
        System.out.println("\n-----\n");
    }
}
```

Add class file 1 into this package

```
package mypackages;
```

```
public class MyResidensialAddress {
    String myResidensialAddress = "No 163 Pillayar Koil Street, \nCrosscut Road, \nMadurai 641538";
    public void DisplayAddress()
    {
        System.out.println("\n-----\n");
        System.out.println("\nFrom mypackages Class 1 :\nMy Residential Address is \n" +
myResidensialAddress);
        System.out.println("\n-----\n");
    }
}
```

Output :

From mypackages Class 1 :
My Residential Address is
No 163 Pillayar Koil Street,
Crosscut Road,
Madurai 641538

From mypackages Class 2 :7
My Office Address is
No 566, Indian Overseas Bank
Airport Campus,,
Chennai 641538

12. Java Threads, using two different threads to print odd and even number.

Objective : Understanding threads, by running block of instructions independently.

Program :

```
import java.util.logging.Level;
import java.util.logging.Logger;
public class JavaProgramsAll {
    public static void main(String[] args) {
        //Threads . . .
        System.out.println("Program for threading");
        MyThreadOdd myThreadOdd = new MyThreadOdd();
        myThreadOdd.start();
        MyThreadEven myThreadEven = new MyThreadEven();
        myThreadEven.start();
    }
}
class MyThreadOdd extends Thread {
    @Override
    public void run() {
        super.run(); //To change body of generated methods, choose Tools | Templates.
        try {
            for(int i = 1; i < 11 ;i++)
            {
                if(i%2 == 1)
                    System.out.println("Running Thread ONE Printing only Odd Number " +i);
                Thread.sleep(2000);
            }
        } catch (InterruptedException ex) {
            System.out.println("Error occurred" + ex.toString());
        }
    }
}
class MyThreadEven extends Thread {
    @Override
    public void run() {
        super.run(); //To change body of generated methods, choose Tools | Templates.
        try {
            for(int i = 1; i < 11 ;i++)
            {
                if(i%2 == 0)
                    System.out.println("Running Thread TWO Printing only even Number " +i);
                Thread.sleep(2000);
            }
        }
    }
}
```

```
    }  
    }  
    catch (InterruptedException ex) {  
        System.out.println("Error occurred" + ex.toString());  
    }  
}  
}
```

Output :

Program for threading

Running Thread ONE Printing only Odd Number 1
Running Thread TWO Printing only even Number 2
Running Thread ONE Printing only Odd Number 3
Running Thread TWO Printing only even Number 4
Running Thread ONE Printing only Odd Number 5
Running Thread TWO Printing only even Number 6
Running Thread ONE Printing only Odd Number 7
Running Thread TWO Printing only even Number 8
Running Thread ONE Printing only Odd Number 9
Running Thread TWO Printing only even Number 10

13. Single thread class multiple task

Objective : understanding threads by sending values and creating multiple threads with different parameters.

Program :

```
import java.util.logging.Level;
import java.util.logging.Logger;

public class JavaProgramsAll {
    public static void main(String[] args) {
        //Threads Multiple with parameters . . .

        System.out.println("Program for multi threading with passing different parameters");
        System.out.println("Printing numbers upto n numbers and by increment (two values as input)");
        System.out.println("-----");
        new MyThread(29, 7).start();
        new MyThread(60, 6).start();
        new MyThread(72, 13).start();
    }
}

class MyThread extends Thread {
    int limit, increment;
    public MyThread(int limit, int increment)
    {
        this.limit = limit;
        this.increment = increment;
    }
    public void run() {
        super.run(); //To change body of generated methods, choose Tools | Templates.
        try {
            for(int i = 1; i <= limit; i = i + increment)
            {
                System.out.println("This is thread for display upto " + limit + " and increment by " +
increment +" : " + i );
                Thread.sleep(2000);
            }
        } catch (InterruptedException ex) {
            System.out.println("Error occurred" + ex.toString());
        }
    }
}
```

Output :

Program for multi threading with passing different parameters
Printing numbers upto n numbers and by increment (two values as input

This is thread for display upto 29 and increment by 7 : 1
This is thread for display upto 60 and increment by 6 : 1
This is thread for display upto 72 and increment by 13 : 1
This is thread for display upto 29 and increment by 7 : 8
This is thread for display upto 72 and increment by 13 : 14
This is thread for display upto 60 and increment by 6 : 7
This is thread for display upto 29 and increment by 7 : 15
This is thread for display upto 72 and increment by 13 : 27
This is thread for display upto 60 and increment by 6 : 13
This is thread for display upto 29 and increment by 7 : 22
This is thread for display upto 72 and increment by 13 : 40
This is thread for display upto 60 and increment by 6 : 19
This is thread for display upto 29 and increment by 7 : 29
This is thread for display upto 72 and increment by 13 : 53
This is thread for display upto 60 and increment by 6 : 25
This is thread for display upto 60 and increment by 6 : 31
This is thread for display upto 72 and increment by 13 : 66
This is thread for display upto 60 and increment by 6 : 37
This is thread for display upto 60 and increment by 6 : 43
This is thread for display upto 60 and increment by 6 : 49
This is thread for display upto 60 and increment by 6 : 55

14. Virtual function

Objective : understanding virtual function - creating a base class function and override base class function in derived class

Program :

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class JavaProgramsAll {
    public static void main(String[] args) {
        System.out.println(" Java - Virtual Function \n -----");
        Base b = new Derived();
        b.show();
    }
}
class Base {
    // virtual by default
    public void show() {
        System.out.println("Base Class function ::show() called");
    }
}
class Derived extends Base {
    public void show() {
        System.out.println("Derived class function ::show() called");
    }
}
```

Output :

Java - Virtual Function

Derived class function ::show() called

15. Exception Handling, Monitoring catching errors or events or any changes and redirection the execution

Objective : Understanding the Exception handling mechanism in java. catching errors or events or any changes, Adding two numbers by input and catch different exceptions

Program :

```
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
public class JavaProgramsAll {
    public static void main(String[] args) {
        //Exception handling, adding two numbers . . . .
        Scanner kinput = new Scanner(System.in);
        String number1, number2;
        System.out.println("Program for Exception Handling");
        System.out.println("-----");
        System.out.println("Adding two integers numbers, "
            + "    \n give input different inputs"
            + "\n Ex float , character, etc");
        try
        {
            System.out.print("\nEnter the first number : ");
            number1 = kinput.next();
            System.out.print("\nEnter the second number : ");
            number2 = kinput.next();
            int result = (Integer.valueOf(number1) + Integer.valueOf(number2));
            System.out.println("\nAddition of " + number1 + " and " + number2 + " is " + result);
        }
        catch(Exception ex)
        {
            System.out.println("Input Error caught is : " + ex.toString());
        }
    }
}
```

Input :

Program for Exception Handling

```
-----
Adding two integers numbers,
give input different inputs
Ex float , character, etc
Enter the first number : 2g
Enter the second number : t6
```

Output :

Input Error caught is : java.lang.NumberFormatException: For input string: "2g"

16. User defined exception, finding not a number and throw exception

Objective : Understanding user define Exception in java. Catching customized exception

Program :

```
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

public class JavaProgramsAll {
    public static void main(String[] args) {
        //User Defined Exception handling, checking the input is number or not . . . .
        Scanner kinput = new Scanner(System.in);
        System.out.println("Program for user defined Exception : ");
        System.out.println("Checking input is a number, if not integer raise exception : ");
        System.out.println("\nInput a number : ");
        if(kinput.hasNextInt())
        {
            System.out.println(" Your input " + kinput.nextInt() +" is a number ");
        }
        else
        {
            String temp = kinput.next();
            try { throw new MyException(temp); }
            catch(Exception ex) { System.out.println("Exception Caught : " + ex.toString());
            }
        }
    }
}

class MyException extends Exception
{
    public MyException(String number ) {
        System.out.println(" Your input " + number +" is not a number ");
    }
}
```

Output :

Program for user defined Exception :
 Checking input is a number, if not integer raise exception :

Input a number :

fg

Your input fg is not a number

Exception Caught : javaprogramsall.MyException

17. I/O in Java (Reading a text file) using scanner class

Objective : Reading a text file using Scanner class

Program :

```
public class JavaProgramsAll {
    public static void main(String[] args) {

        System.out.println("Reading a text file and display the content");
        try {
            Scanner scanner = new Scanner(new File("f:\\myprofile.txt"));
            while (scanner.hasNextLine())
            {
                System.out.println(scanner.nextLine());
            }
            scanner.close();
        } catch (Exception ex)
        {
            System.out.println("Error occurred : " + ex.toString());
        }
    }
}
```

Output :

Reading a text file and display the content

My Profile

My Profile

Name : S.Kumar
Date of Birth : 27.10.1990
Address : No 183, Kamaraj Street
Madurai

18. I/O in Java - Writing into a text file using FileWriter class

Objective : understanding to write the string content into a text file using FileWriter class

Program :

```
import java.io.FileWriter;
import java.io.IOException;

public class JavaProgramsAll {
    public static void main(String[] args) throws IOException {
        String title = "This is the file content , \ngoing to write into file\n\n ";
        String myaddress = "My Address \n ----- S.Kumar \n No 566 Kumaran Street \n
Madurai 641589";
        FileWriter fileWriter = new FileWriter("f:\\myaddress.txt");
        for (int i = 0; i < myaddress.length(); i++)
            fileWriter.write(myaddress.charAt(i));
        System.out.println("Content Updated into text file : myaddress.txt ");
        fileWriter.close();
    }
}
```

Output :

Content Updated into text file : myaddress.txt

19. I/O in Java - Writing into a text file using FileWriter class

Objective : understanding to write the string content into a text file using FileWriter class

Program :

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class JavaProgramsAll {
    public static void main(String[] args) throws IOException {
        int onechar;
        FileReader fileReader = null;
        fileReader = new FileReader("f:\\myaddress.txt");
        while ((onechar=fileReader.read())!=-1)
            System.out.print((char)onechar);
        fileReader.close();
    }
}
```

Output :

My Address

S.Kumar

No 566 Kumaran Street

Madurai 641589

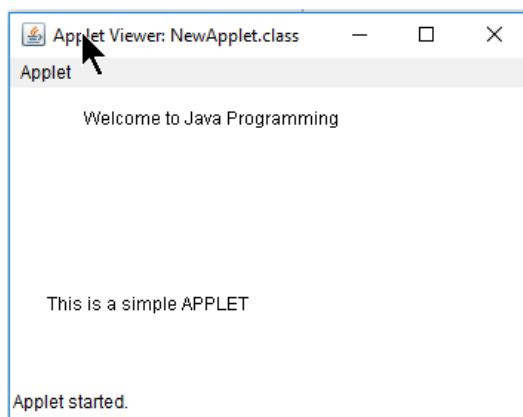
20. Java Applet – A simple program using NETBEANS IDE

Objective : Understanding applet in java, using NETBEANS IDE

Program : Create a applet in NETBEANS IDE and run the file

```
import java.applet.Applet;
import java.awt.Graphics;
public class NewApplet extends Applet {
    public void init() {
    }
    public void paint(Graphics g) {
        g.drawString("Hello applet!", 50, 25);
    }
}
```

Output :



21. Applet – Embedding with html

Objective : Understanding to run the java applet inside the web browser, and embed in into html code

Program :

Save the following java applet program as GraphicsDemo.java, and compile it, save the compiled class file and html file in the same directory

```
import java.applet.Applet;
import java.awt.*;

public class GraphicsDemo extends Applet{
    public void paint(Graphics g){
        g.setColor(Color.red);
        g.drawString("Welcome",50, 50);
        g.drawLine(20,30,20,300);
        g.drawRect(70,100,30,30);
        g.fillRect(170,100,30,30);
        g.drawOval(70,200,30,30);
        g.setColor(Color.pink);
        g.fillOval(170,200,30,30);
        g.drawArc(90,150,30,30,30,270);
        g.fillArc(270,150,30,30,0,180);
    }
}
```

Save the following code as myapplet.html

```
<html>
  <body>
    <applet code="GraphicsDemo.class" width="300" height="300">
  </applet>
</body>
</html>
```

22. Networking – Retrieving the current system IP Address

Objective : Getting current System IP Address

Program :

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class JavaProgramsAll {
    public static void main(String[] args) {
        try {
            InetAddress ipAddr = InetAddress.getLocalHost();
            System.out.println("This System ip address is : \n " + ipAddr.getHostAddress());
        } catch (UnknownHostException ex) {
            ex.printStackTrace();
        }
    }
}
```

Output :

This System ip address is :
186.172.145.190

23. Networking – Retrieving the host system IP Address

Objective : Retrieving the host system ip address from the given website name

Program :

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class JavaProgramsAll {
    public static void main(String[] args) {
        try {
            InetAddress host = InetAddress.getByName("www.w3schools.com");
            System.out.println("IP Address for www.w3schools.com is : " + host.getHostAddress());
        } catch (UnknownHostException ex) {
            ex.printStackTrace();
        }
    }
}
```

Output :

IP Address for www.w3schools.com is : 192.229.179.87

24. Networking – Connection establishment between Client and Server

Objective : Establish the connection between a client and a server, then send a message from client to server

Program :

MyServer.java

```
import java.io.*;
import java.net.*;

public class MyServer {
public static void main(String[] args){
try{
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept();//establishes connection

DataInputStream dis=new DataInputStream(s.getInputStream());

String str=(String)dis.readUTF();
System.out.println("Connected with Client... \n Received message \n" + str);

ss.close();

}catch(Exception e){System.out.println(e);}
}
}
```

MyClient.java

```
import java.io.*;
import java.net.*;

public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);

DataOutputStream dout=new DataOutputStream(s.getOutputStream());
```

```
dout.writeUTF("Hi, This is Client System . . . \n Hi Good Morning . . .");  
dout.flush();
```

```
dout.close();  
s.close();
```

```
}catch(Exception e){System.out.println(e);}  
}  
}
```

Output at server system :

Connected with Client...
Received message

Hi, This is Client System . . .
Hi Good Morning . . .

25. Networking – Chat text between Client and Server

Objective : Developing a text chat conversation between two systems, Two java programs running MyServer java program at one system and MyClient java program at another system.

Program :

MyServer.java

```
import java.net.*;
import java.io.*;
class MyServer{
public static void main(String args[])throws Exception{
ServerSocket ss=new ServerSocket(3333);
Socket s=ss.accept();
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("stop")){
str=din.readUTF();
System.out.println("client says: "+str);
str2=br.readLine();
dout.writeUTF(str2);
dout.flush();
}
din.close();
s.close();
ss.close();
}}
```

MyClient.java

```
import java.net.*;
import java.io.*;
class MyClient{
public static void main(String args[])throws Exception{
Socket s=new Socket("localhost",3333);
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
```

```
while(!str.equals("stop")){
str=br.readLine();
dout.writeUTF(str);
dout.flush();
str2=din.readUTF();
System.out.println("Server says: "+str2);
}

dout.close();
s.close();
}}
```

Output :

Output window from

MyServer Java Program

client says: Hi Good Morning
Hello, how are yoou
client says: I am fine
ok good bye
client says: ok bye
Stop

Output window from

MyClient Java Program

Hi Good Morning
Server says: Hello, how are yoou
I am fine
Server says: ok good bye
ok bye
Server says: Stop