

Programming in C

Topic : Basic Programs

1. Write a program to print Hello World on output screen.
2. WAP to calculate Simple interest.
3. WAP to find out perimeter and area of the Square and Rectangle.
4. WAP to convert Fahrenheit temp in degree Celsius.
5. WAP to demonstrate arithmetic operation on two integer numbers.
6. Write a program which will accept three integer numbers from user and find out greatest among them using conditional operator.
7. Accept two numbers and perform swapping with third variable, and without third variable
[Infosys]
8. WAP to accept five subject marks and find out total and average of the marks.
9. WAP which calculate speed for time and distance.
 $\text{Speed} = (\text{distance} / \text{time})$
10. Command line program to find area of triangle **[Amdocs]**

Topic : Selection statement

1. WAP to check whether a number is even or odd
2. WAP to check whether a person is eligible for voting or not.
3. WAP to accept Cost Price from user and ask whether the user is a student or not. If the user is student and cost price is greater than 500, give discount of 10% ELSE discount will be 5%. If user is not student and cost price is greater 500 then give discount of 8% ELSE discount will be 2%.
(Take all inputs from USER)
4. WAP to check whether Number is positive or negative or ZERO.
5. WAP a program to accept Percentage from user and check the GRADE
 - A. Above 70% - Grade A
 - B. Between 60% to 70% - Grade B+.
 - C. Between 45% to 60% - Grade B.
 - D. Between 35% to 45% - Grade C.
 - E. Less than 35% - Fail

Lab Manual

6. Accept three numbers from user and find out largest number among three and also find out whether that three numbers are equal or not. (else if ladder)

Topic : Branching statement

1. WAP using following menus-

Choice-1: Accept number and find out square and cube.

Choice-2: Check whether the given year is LEAP or not. *[TCS]*

If user enters wrong choice appropriate message should get displayed.

2. WAP using switch case for arithmetic operation on two numbers, if user enters an operator as choice, the appropriate operation should perform.

If user enters wrong choice appropriate message should get displayed.

i.e. + is for addition

- is for subtraction

Topic : Looping statements

1. WAP to check whether a number is prime or not. *[TCS]*

2. WAP to accept a number from user and find out sum of even digits from that given number.

3. WAP to print the following pattern : *[MindTree]*

```
      *
     ***
    *****
   ********
  *********
 *****
```

4. WAP to print the following on output screen using jumping statements

```
1    5
2    4
4    2
5    1
```

5. WAP to print *[AMCAT, Wipro]*

```
1 1 1 2
3 2 2 2
3 3 3 4
```

Lab Manual

6. GCD of three numbers [*Wipro,TCS*]
7. Find all pythagorean triplet below the given limit. [*Cognizant*]
8. WAP to print [*AMCAT exam*]

```
1
1 1
1 0 1
1 0 0 1
1 1 1 1 1
```

9. WAP to convert decimal to binary /binary to octal [*CoCubes/Amcat*]

Topic : Functions

1. WAP to demonstrate all four categories of functions for volume of the cylinder
(volume of cylinder: $3.14 * r * r * h$)
 - a) Function without parameters without return value.
 - b) Function with parameter without return value.
 - c) Function without parameter with return value.
 - d) Function with parameters with return value.
2. WAP to calculate factorial of a number using a function. (using recursion)
[*TCS,Persistent,Infosys*]
3. WAP to print Fibonacci series.(also using recursion) [*TCS*]
4. WAP to check whether a number is Armstrong number. [*TCS*]
5. WAP to check whether a number is palindrome. [*TCS*]
6. WAP to find HCF and LCM of given number using recursion. [*IBM*]

Topic : Pointers

1. Write a program that declares a double, an int, and char variables. Next declare and initialize a pointer to each of the three variables. Your program should then print the address of, and value stored in.
2. Demonstrate addition of two floating type numbers by using call by address.

Lab Manual

3. Demonstrate pointer arithmetic by assigning pointer “ptr” to variable “a” which is having value as 100 in it. Perform increment operation on pointer ptr like ptr=ptr+2 and display which address that pointer ptr hold.

Topic : Array

1. WAP to calculate average marks of a 10 students.
2. WAP to sort array in ascending order.
3. WAP to find greatest and smallest number in an array.
4. WAP to find transpose of matrix *[Infosys]*
5. WAP to multiply a 3*3 matrix. *[TCS, Infosys]*
6. Write a C program to compute sum of diagonal elements of an array *[Infosys]*
7. Write a program to search particular value in an array and return the index number where it is located.
8. Perform following operations on two matrices with order m*n and p*q.
9. Addition of two matrix
10. Multiplication of two matrix
11. WAP to check whether two matrices are identical or not. *[Cognizant]*
12. Given a 2D array, print it in spiral form. *[Cognizant]*

I/p 1 2 3 4 5
 6 7 8 9 10
 11 12 13 14 15

O/p 1 2 3 4 5 10 15 14 13 12 11 6 7 8 9

13. Write a program to print all the LEADERS in the array. (An element is LEADER if it is greater than all the elements to its right side.) *[CoCubes]*
14. Sort array using bubble sort *[Persistent]*

Topic : String

1. WAP to compare two strings using strcmp ().
2. WAP to concatenate two strings without using library function.
3. WAP to convert upper-case string into lower-case and vice versa .Write your own functions for the same. *[TCS]*

Lab Manual

4. Find a character(or sub-string) in a string without using library function. And print its ascii value **[IBM]**
5. Eliminate vowels from string **[MindTree]**
6. WAP to check a string is palindrome.(Example Wow,bob,radar,sagas..etc). **[TCS]**
7. WAP to count no of blank spaces in your paragraph without using string function and write it in your own function.
8. WAP to reverse the sentence or string without using library function.**[MindTree,TCS]**

for example

“hi all”

reverse 1: “all hi”

reverse 2: “lla ih”

9. WAP to show all possible library function for reading and writing String .
10. WAP to convert alternate characters of the string to uppercase.The first letter of the string has to be capital. **[IBM]**
I/p: We are the world
O/p: We ArE tHe WoRLD
11. WAP to transform string of lowercase character by replacing each letter with subsequent character. **[IBM]**
12. write a program to count different types of characters in given string. **[Infosys]**
13. Dynamically read a string and sort it using bubble sort **[Persistent]**

Topic : Dynamic Memory Allocation

1. Write a program to create memory for int, char and float variable at run time.
2. Write a program to input and print text using Dynamic Memory Allocation.
3. Write a program to read a one dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.
4. Write a program to read and print the student details using structure and Dynamic Memory Allocation.
5. Write a program to read and print the N student details using structure and Dynamic Memory Allocation.

Lab Manual

Topic : File Handling

1. Write a program to read name and marks of n number of students from user and store them in a file.
2. Write a program to read name and marks of n number of students from user and store them in a file. If the file previously exists, add the information of n students.
3. Write a program to write all the members of an array of structures to a file using fprintf(). Read the array from the file and display on the screen.
4. Write to a text file using fprintf()
5. Write a Program to Append the Content of File at the end of Another.
6. Write a Program to Capitalize First Letter of every Word in a File **[IBM]**
7. Write a Program to Count No of Lines, Blank Lines, Comments in a para in a file .
8. Write a c program which produces its own source code as its output. **[Persistent]**
9. Write a program to remove spaces from a File and store the contents without space in a new file. **[Persistent]**

Topic : Structure and Union

1. Create a structure Student. Create array of 10 students using MACRO and display data in tabular form.
2. Create a structure Employee. Accept data for 5 employees and display it.
3. Demonstrate use of union and find out size of union variable and the attributes from union.
4. WAP to demonstrate typedef keyword.
5. WAP to copy one structure into another. Use concept of nested structures.
6. Create Employee structure having attributes as id, name and basic salary.
7. Create Date structure having attributes as dd,mm,yy.
8. Nest Date structure into Employee as to display joining date of employee.
9. Create a structure Employee. Pass it to a function by value.
10. Make a structure for banking customer which includes all the essential information like Name,Acc. No, Balance etc **[Persistent]**

OOPs With C++

Topic : CLASS AND OBJECTS

1. Create the following classes and write their appropriate class members and display proper information to user.

a) Time b) Date c) Person d) Student e) Fan f) Point g)Box

Create a class Team as follows with following data members such as, Country_Name, name of Player, age, no_of_matches, batting_avg, balling_avg;

Accept 5 different records in array and display the records as follows:

Country Name	Player Name	Matches	Age	BattingAvg	BallingAvg
India	Sachin	295	30	45.51	53.00
Australia	Ricky	160	28	41.00	67.00
India	Saurav	230	31	40.95	30.00

2. An electricity board charges the following rates to domestic users to discourage large consumption of energy:

For the first 30 units - 1.50/- per unit

For next 200 units - 3.00/- per unit

Beyond 300 units - 4.25/- per unit

Create a class ElectricityBill for atleast 5 users with following operations....

Write a program to read the name of the user and number of units consumed and print out the charges with names.

If the total amount is more than Rs. 500.00 then an additional rcharge of 15% is added.

3. Write a program to implement a sphere class with appropriate members and member function to find the surface area and volume: (Surface = $4\pi r^2$ and vol = $\frac{4}{3}\pi r^3$)

4. Write a program to implement a telephone bill class with Name, Address, Tel. No., No. of calls as data members. Compute the amount to be paid if the charges per call is Rs. 2/-.

Topic : Constructor and Destructor

1. Create the following classes and write member functions and display proper information to user using constructor and destructor.

a) Time b) Date c) Person d) Student e) Fan f) Point

2. Define a class to represent a bank account. Include the following members

Data Members

1. Name of the depositor

Lab Manual

2. Account number
3. Type of account
4. Balance amount in the account.

Member functions

- 1) To assign initial values
- 2) To deposit an amount
- 3) To withdraw an amount after checking the balance
- 4) To display name and balance.

3. Define class student which shows the information about the Student using constructor and destructor - Include the following members

Data Members

1. Name of the Student
2. Roll No
3. Address
4. Percentage

Member functions

1. To assign initial values
2. To accept the values from user
3. To display all data to user including Grade by using following conditions
 - a. $\text{Per} \geq 70$ Distinction
 - b. $\text{Per} < 70$ and $\text{Per} \geq 60$ First Class
 - c. $\text{Per} < 60$ and $\text{Per} \geq 40$ Second class
 - d. $\text{Per} < 40$ Student is Fail

Topic : Reference variable, Copy Contructor

1. Create the following classes and write member functions and display proper information to user using constructor and destructor also use this pointer.
a) Time b) Date c) Person d) Student e) Fan f) Point
2. Swap two numbers using reference variable. Create class Swap.
3. Write a program to implement flight class with data member as flight no., source, destination and fare. Write a copy constructor and a member function to display the flight information.
4. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and no. of copies. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the

Lab Manual

requested copies is displayed; otherwise the message “Required copies not in stock” is displayed.

Incorporate the following features:

a. Value of attributes should be assign to some value.

b. Create Accept_book_Info() and Display_Book_Status() methods.

5. Create one class Library which having data members as author, title, price, publisher and no. of copies. Assign some specific values to all its data members and create member functions as Accept_Book_info(), Print_Book_info().

Create function such as Selling_Prise() on the following basis of conditions:

If Actual price of the book is greater than equal to 2500/- then discount=1.00%

If actual price is less than 2500/- then discount=0.5%

Then find out final selling price of the book.

Topic : Inline, friend function and static data members

1. Implement inline function for the following programs-
 - a. Square of the number
 - b. Cube of the number
 - c. Area of rectangle
2. Find out largest number between two numbers by implementing two different classes as-
 - a. Number1-include n1 as variable and get_a(), disp_a() as functions.
 - b. Number2-include n2 as variable and get_b(), disp_b() as functions.
 - c. Create one common function as friend of both Number1 and Number2 class.
3. Demonstrate static member function with static data member in one class.

Topic : Function overloading

1. Demonstrate function overloading for addition of two integer and two floating point numbers.
2. Swap two integer numbers and two characters using function overloading.
3. Area of Rectangle, Circle and Square.

Topic : Operator overloading

1. Demonstrate unary ++ operator using operator function
2. Make addition of Two Complex numbers using binary + operator(using friend function)
 - a. C1- 20+i30

Lab Manual

- b. C2- 14+i10
 - c. Resultant complex number is- 34+i40
- 3. Add two values times and display resultant time in the hr.: min: sec form using friend function(overload binary + operator)
 - a. T1- 2:40:35
 - b. T2- 3:35:30

Resultant time is- 06:16:05

Topic : Generic Functions And Exception Handling

1. WAP using generic function to find sum of 2 numbers.
2. WAP using exception handling if in the division the denominator is negative.

Database

Table Name : Employee

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Services
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Services
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurance

Lab Manual

Table Name : Incentives

Employee_ref_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

1. Create Table Employee & Incentives using above table
2. Add Primary key Constrains to both table;
3. Write Sql Syntax to create EMPLOYEE_REF_ID in INCENTIVES table as foreign key with respect to EMPLOYEE_ID in employee table
4. Write SQL to drop foreign key on employee table

Topic : Select

1. Get all employee details from the employee table
2. Get First_Name, Last_Name from employee table
3. Get First_Name from employee table in upper case

Topic : SQL Order By

1. Get unique DEPARTMENT from employee table
2. Get all employee details from the employee table order by First_Name Ascending
3. Get all employee details from the employee table order by First_Name Ascending and Salary descending

Topic : SQL Group By Query

1. Get department, total salary with respect to a department from employee table.
2. Get department, total salary with respect to a department from employee table order by total salary descending
3. Get department, no of employees in a department, total salary with respect to a department from employee table order by total salary descending

Lab Manual

4. Get department wise average salary from employee table order by salary ascending
5. Get department wise maximum salary from employee table order by salary ascending
6. Select no of employees joined with respect to year and month from employee table
7. Select department,total salary with respect to a department from employee table where total salary greater than 800000 order by Total_Salary descending

Topic : SQL Where Condition

1. Get employee details from employee table whose employee name is “John”
2. Get employee details from employee table whose employee name are “John” and “Roy”
3. Get employee details from employee table whose employee name are not “John” and “Roy”

Topic : SQL Wild Card Search

1. Get employee details from employee table whose first name starts with 'J'
2. Get employee details from employee table whose first name contains 'o'
3. Get employee details from employee table whose first name ends with 'n'

Topic : SQL Pattern Matching

1. Get employee details from employee table whose first name ends with 'n' and name contains 4 letters
2. Get employee details from employee table whose first name starts with 'J' and name contains 4 letters
3. Get employee details from employee table whose Salary greater than 600000
4. Get employee details from employee table whose Salary less than 800000
5. Get employee details from employee table whose Salary between 500000 and 800000
6. Get employee details from employee table whose name is 'John' and 'Michael'

Lab Manual

Topic : SQL Join

1. Select first_name, incentive amount from employee and incentives table for those employees who have incentives
2. Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000
3. Select first_name, incentive amount from employee and incentives table for all employees even if they didn't get incentives
4. Select first_name, incentive amount from employee and incentives table for all employees who got incentives using left join

Topic : SQL DATE Functions

1. Get employee details from employee table whose joining year is "2013"
2. Get employee details from employee table whose joining month is "January"
3. Get employee details from employee table who joined after January 31st
4. Get database date
5. Get Employee ID's of those employees who didn't receive incentives without using sub query ?(MINUS)
6. Delete employee data from employee table who got incentives in incentive table
7. Update incentive table where employee name is 'John'

Topic : SQL Union

1. Select First_Name, LAST_NAME from employee table as separate rows

Topic : Views

1. Create view for Employee which displays id name of Employee.
2. Update view to add Salary column to View
3. drop View

Topic : INDEX

Lab Manual

1. Creates an index on a table Employee.
2. Create Forced Index on new Table which is not exit in Database
3. Drop Index

M.A.D.

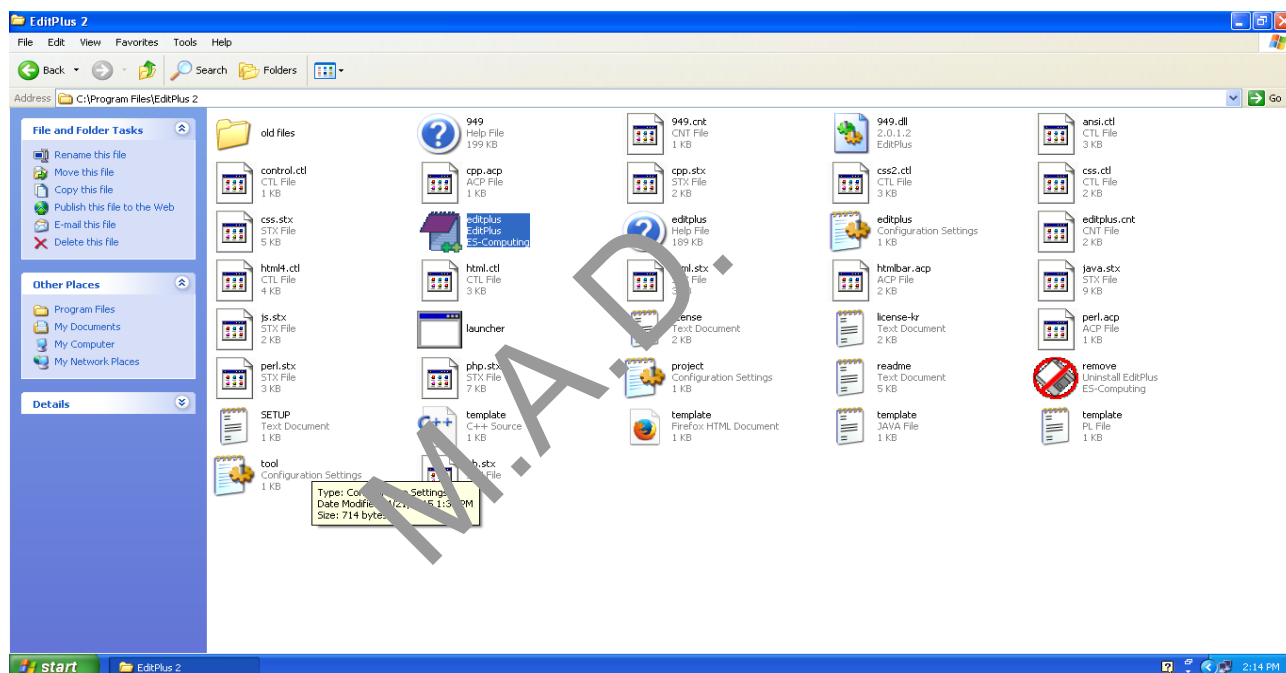
Lab Manual

CORE JAVA

Lab 1

Writing First Java Program

4. Click on start->My Computer-> System (C) -> Program Files -> Edit Plus 2..
5. Double click on editplus icon

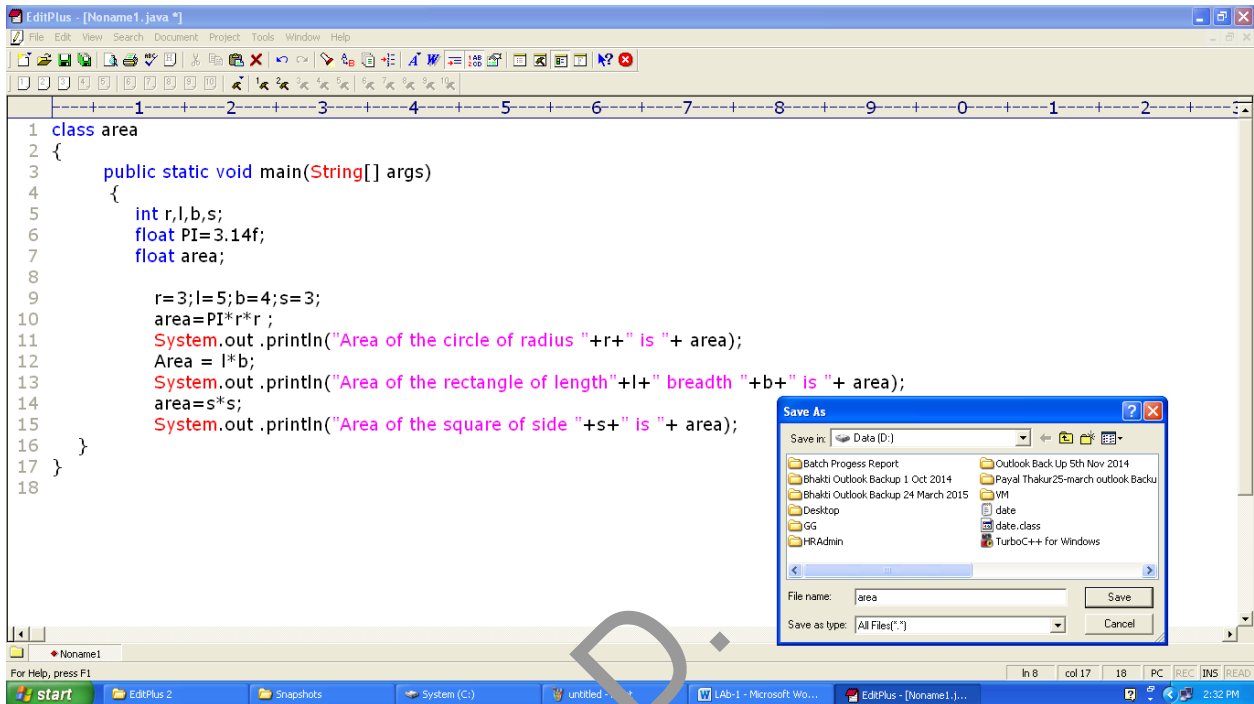


Click on I Agree

File -> New -> Java

Write your java program in the editor

Lab Manual

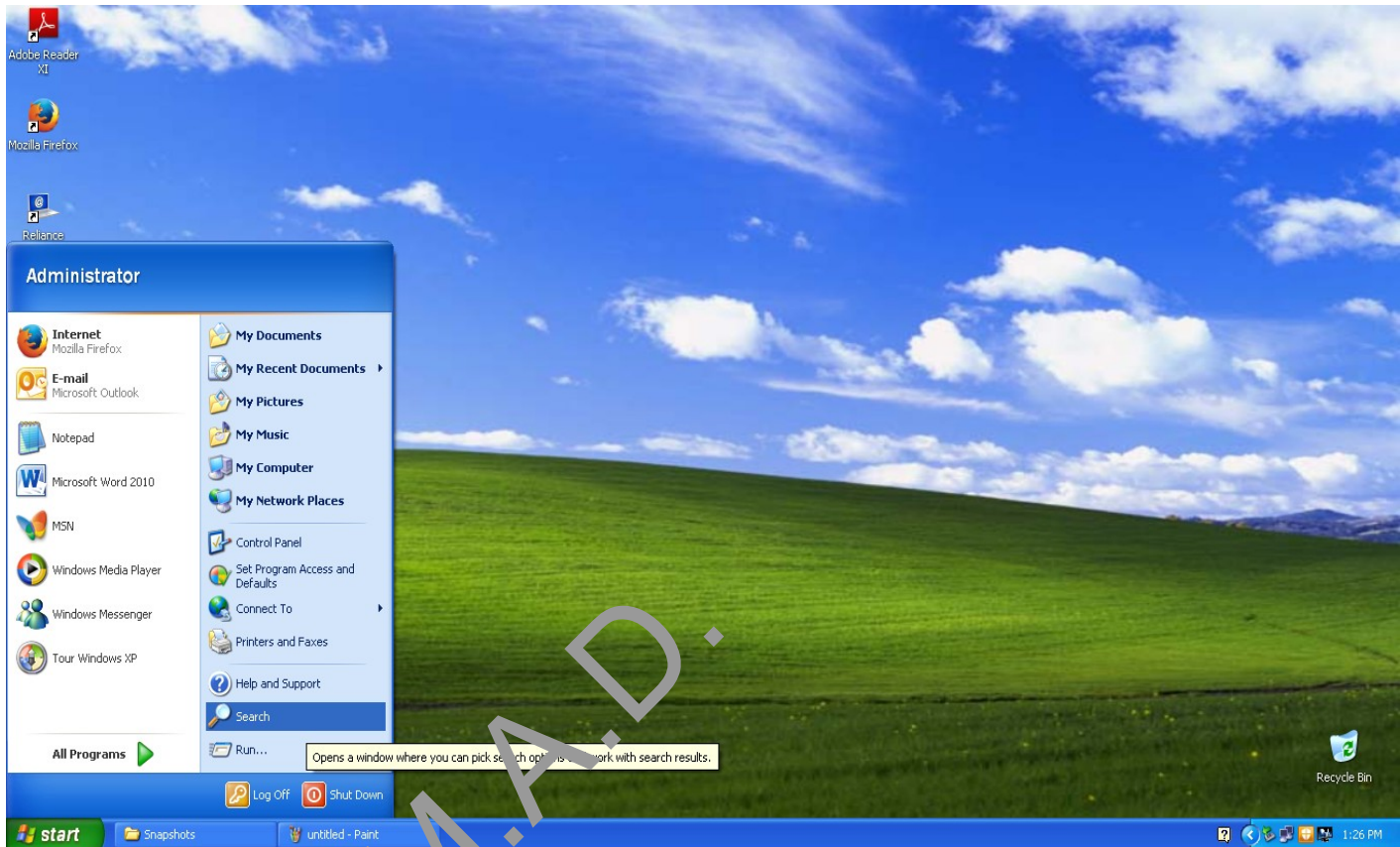


Save the file as area.java (saved in D drive)

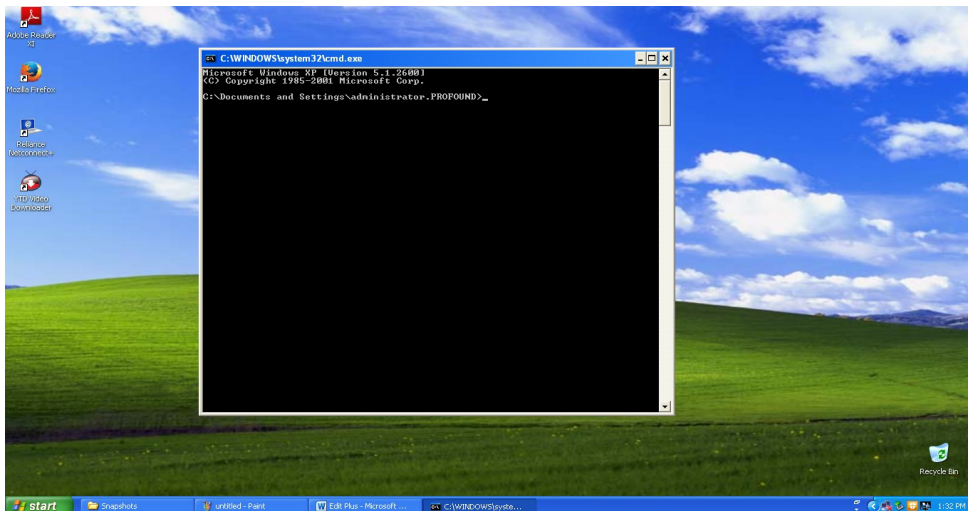
Compile and Run Java Program on Command Prompt Steps:

5. Click on Start

Lab Manual



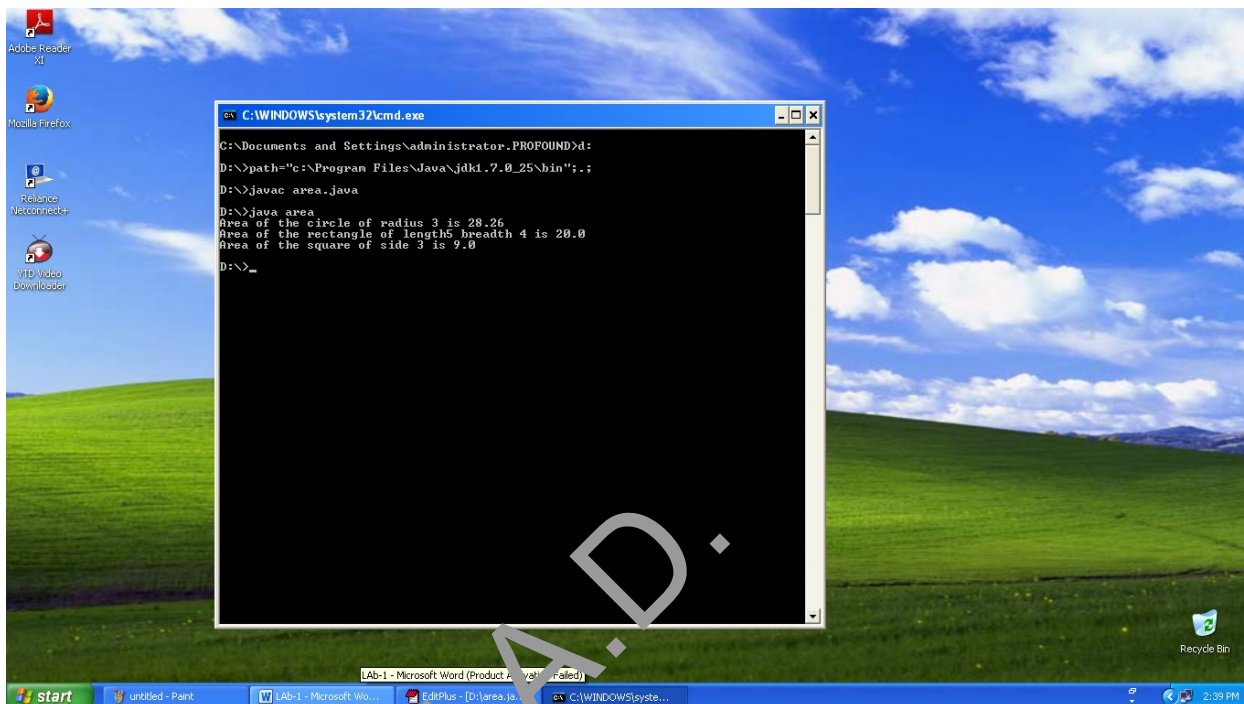
6. Click on run start->run (search cmd)
7. Type cmd and click on OK Button



8. From the command prompt move to the drive and folder where you have saved your program like
`c:\Documents and Settings\administrator.PROFOUND> d:(press Enter Key)`

Lab Manual

d:\>



5: set the path

D:\> path="c:\Program Files\Java\jdk1.7\bin";.; //(Press enter key)

5) Compile the source code

D:\>javac area.java

Once compiled generates area.class file

6) Run the .class file

D:\>java area

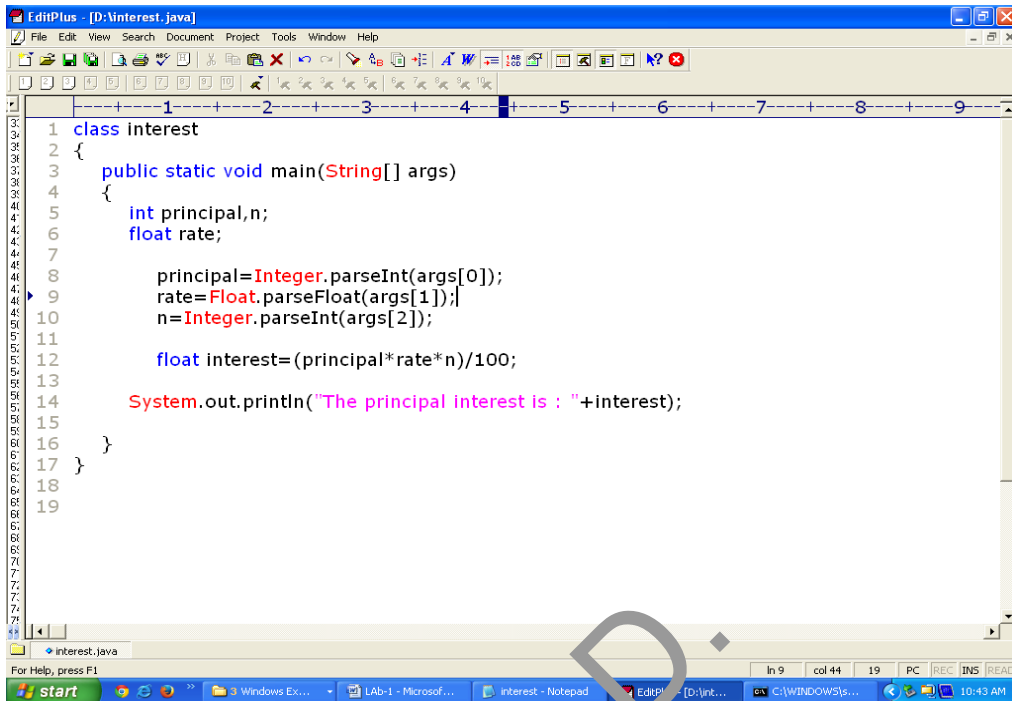
Area of the circle of radius 3 is 28.26

Area of the rectangle of length5 breadth 4 is 20.0

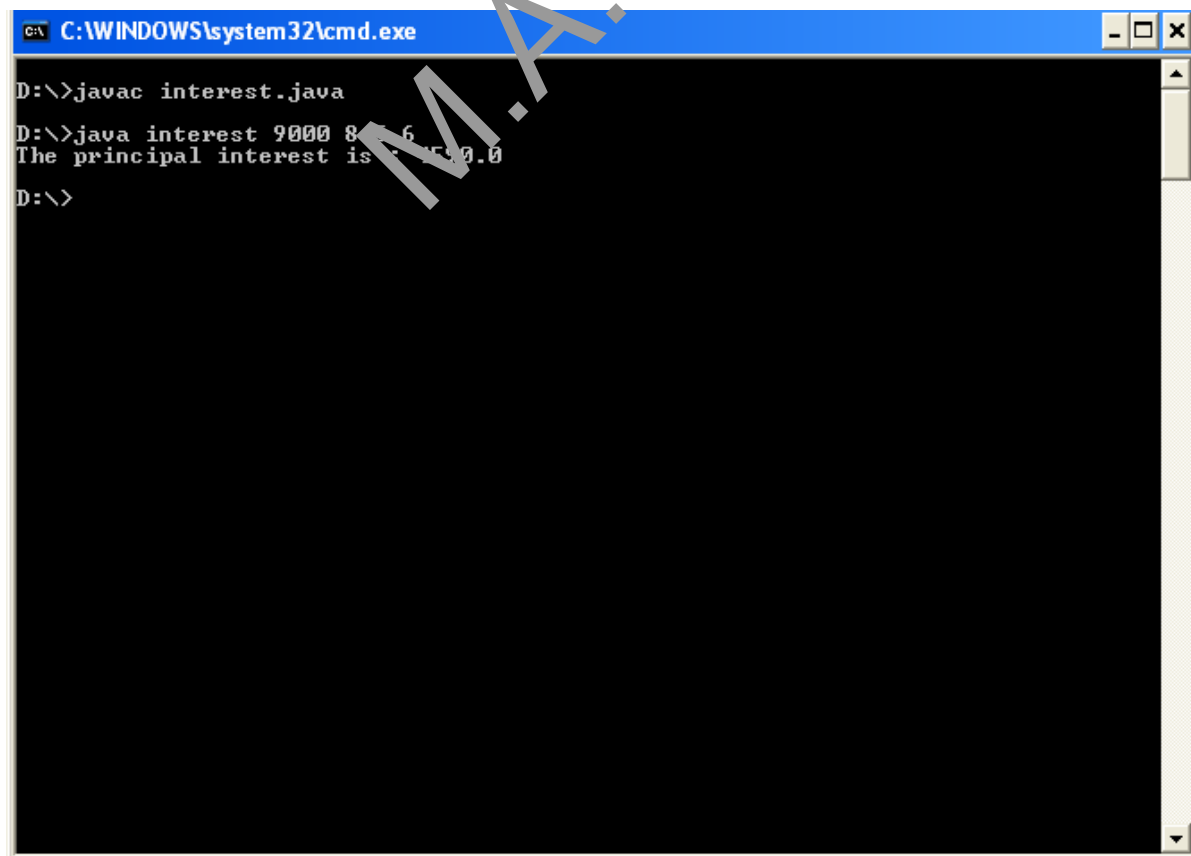
Area of the square of side 3 is 9.0

Demo for Command Line Arguments

Lab Manual



```
1 class interest
2 {
3     public static void main(String[] args)
4     {
5         int principal,n;
6         float rate;
7
8         principal=Integer.parseInt(args[0]);
9         rate=Float.parseFloat(args[1]);
10        n=Integer.parseInt(args[2]);
11
12        float interest=(principal*rate*n)/100;
13
14        System.out.println("The principal interest is : "+interest);
15    }
16 }
17 }
18
19
```



```
C:\WINDOWS\system32\cmd.exe

D:\>javac interest.java

D:\>java interest 9000 8 6
The principal interest is : 480.0

D:\>
```

Assignments To Solve

1. accept two integer values via CLA* and perform all arithmetic operation2

Lab Manual

2. Accept five different values via CLA* by using for loop and display sum of that values
3. Accept values via CLA* for 1-D array of integer type and display it on screen and calculate the sum of number in array
4. Given an array of integers A and a sum B, find all unique combinations in A where the sum is equal to B. **(Infosys)**
Each number in A may only be used once in the combination.

Note:

All numbers will be positive integers.

Elements in a combination (a_1, a_2, \dots, a_k) must be in non-descending order. (ie, $a_1 \leq a_2 \leq \dots \leq a_k$).

The combinations themselves must be sorted in ascending order.

If there is no combination possible the print "Empty" (without quotes).

Example,

Given A = 10,1,2,7,6,1,5 and B(sum) 8,

A solution set is:

[1, 7]

[1, 2, 5]

[2, 6]

[1, 1, 6]

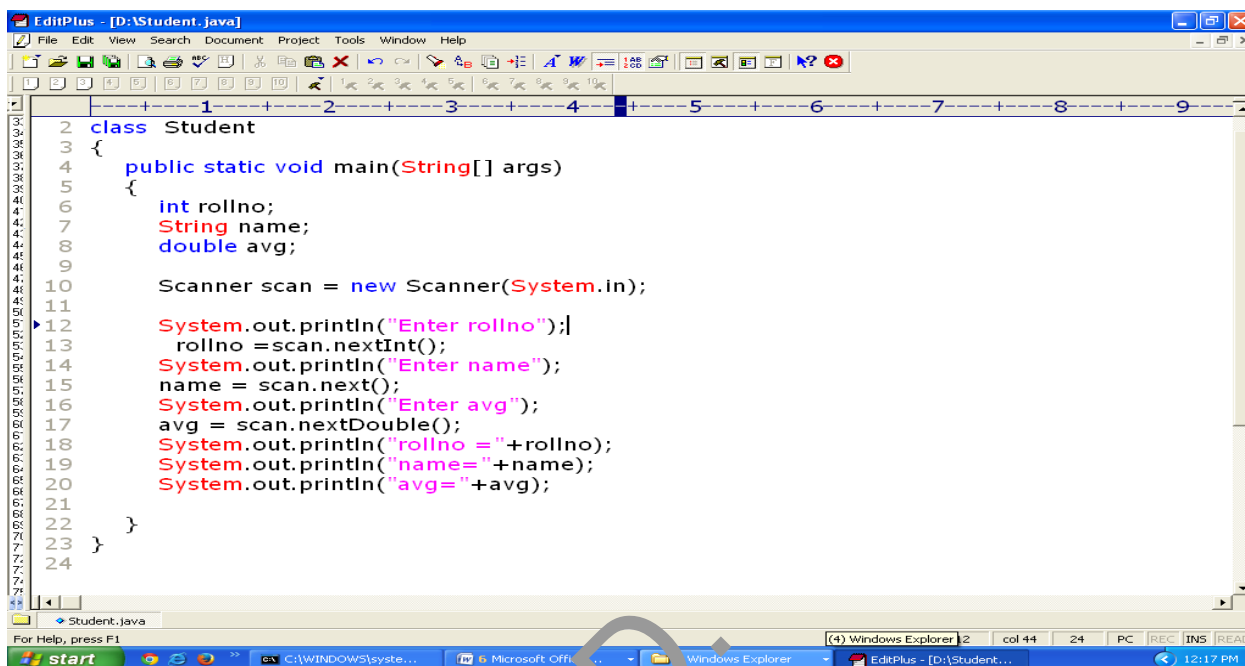
5. Write a java program to check number is perfect or not. **(Accenture)**

Lab 2

Classes and Objects Using Scanner class

1. Introducing Scanner Class To Accept The Values From The User Demo for Scanner Class

Lab Manual

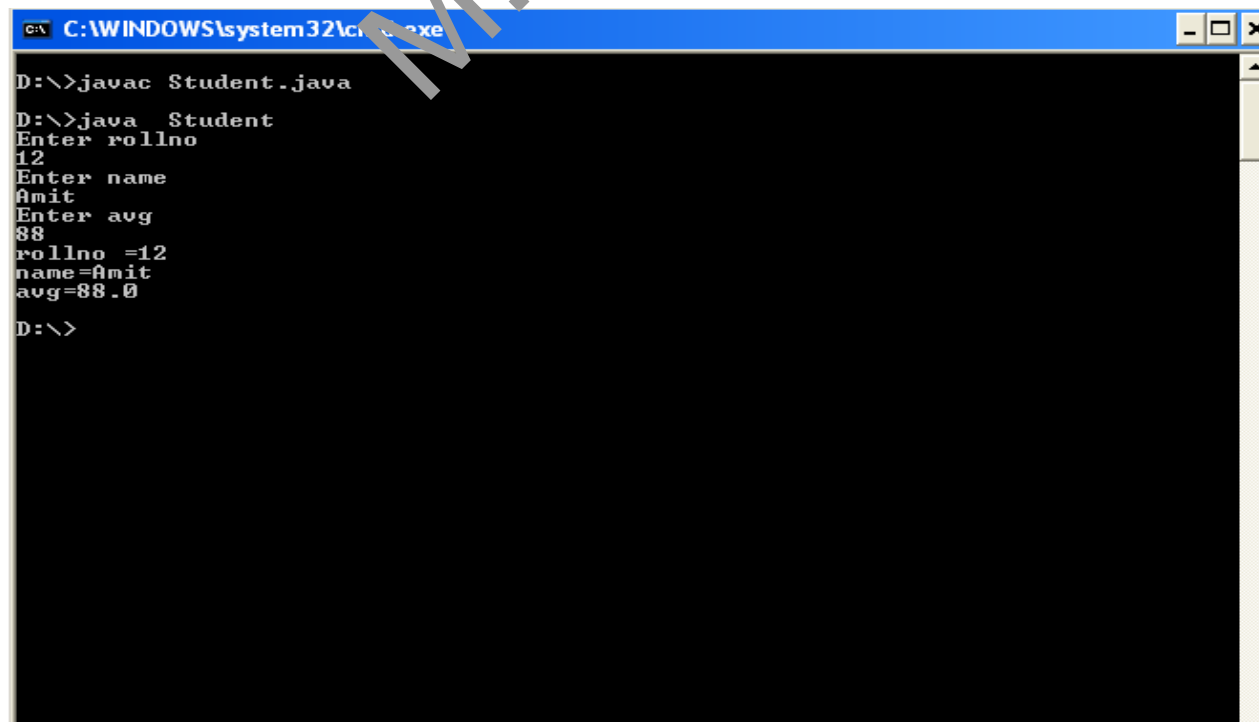


```
1 class Student
2 {
3     public static void main(String[] args)
4     {
5         int rollno;
6         String name;
7         double avg;
8
9         Scanner scan = new Scanner(System.in);
10
11         System.out.println("Enter rollno");
12         rollno = scan.nextInt();
13         System.out.println("Enter name");
14         name = scan.next();
15         System.out.println("Enter avg");
16         avg = scan.nextDouble();
17         System.out.println("rollno =" + rollno);
18         System.out.println("name =" + name);
19         System.out.println("avg =" + avg);
20
21     }
22 }
23
24
```

Save as Student.java(in d:\)

Compile and run from Command Prompt

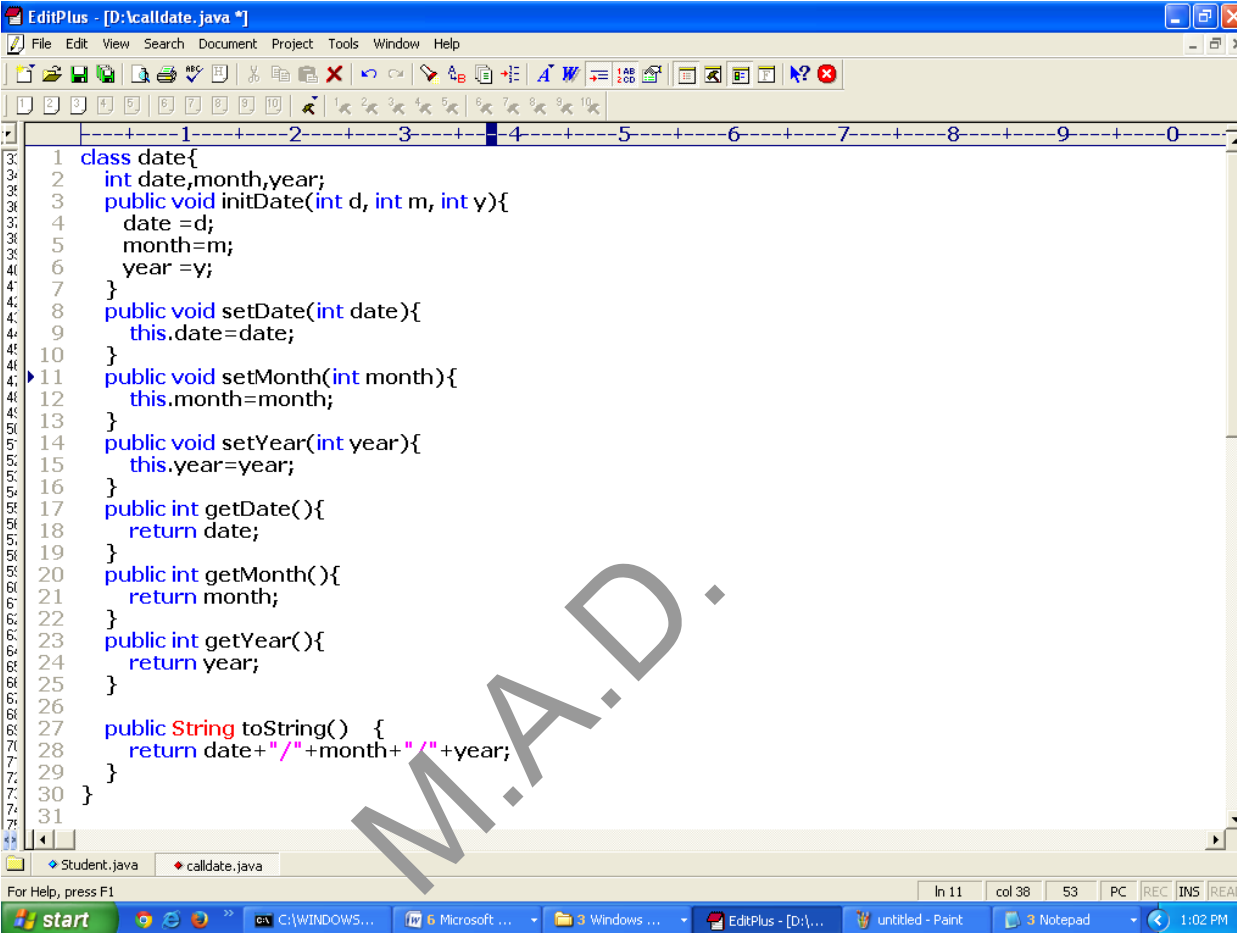
Note: Always set the path if new command prompt is opened(as mentioned in the previous lab demo)



```
C:\WINDOWS\system32\cmd.exe
D:\>javac Student.java
D:\>java Student
Enter rollno
12
Enter name
Amit
Enter avg
88
rollno =12
name=Amit
avg=88.0
D:\>
```

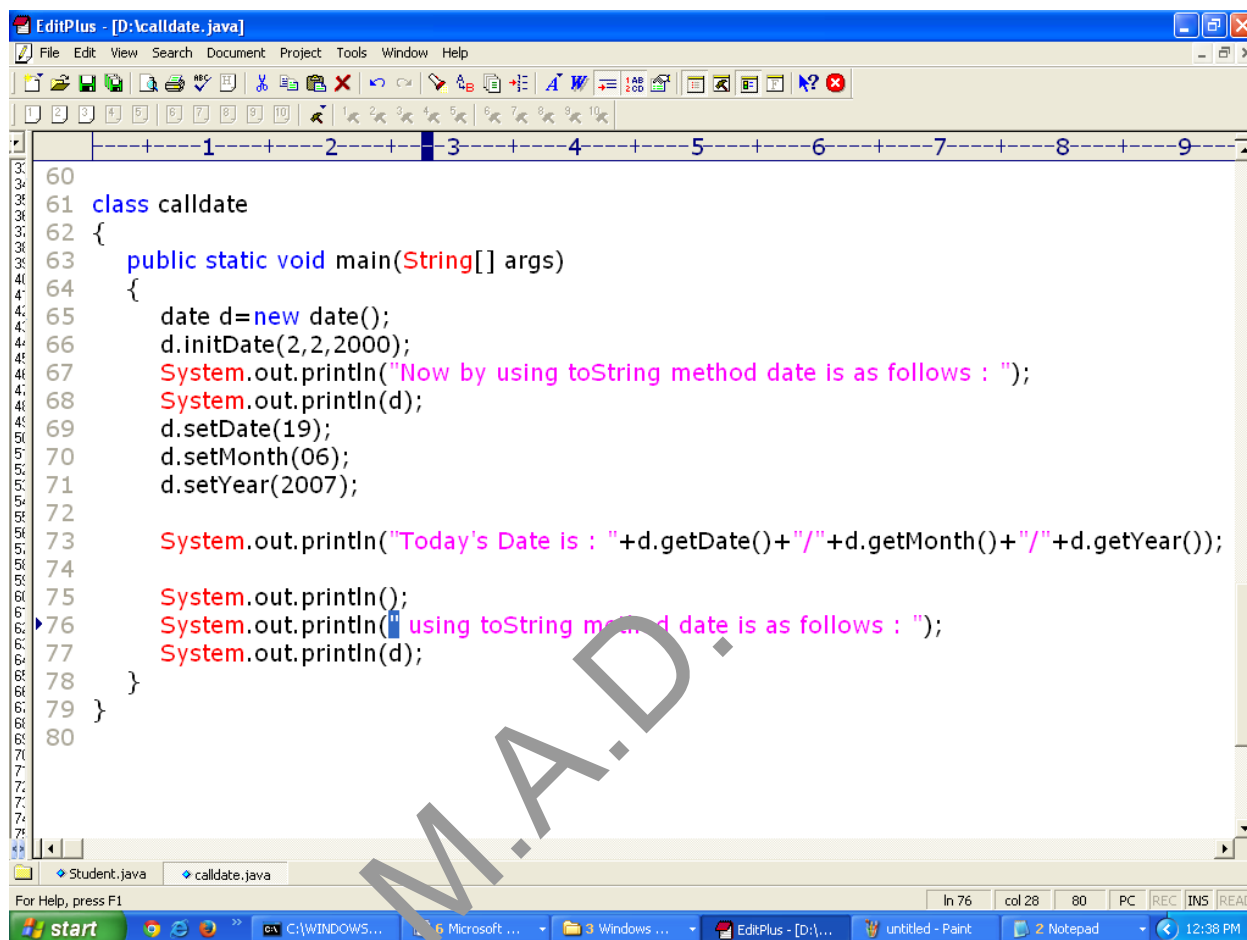
Lab Manual

2.Demo for Getter Setter Methods



```
1 class date{
2     int date,month,year;
3     public void initDate(int d, int m, int y){
4         date =d;
5         month=m;
6         year =y;
7     }
8     public void setDate(int date){
9         this.date=date;
10    }
11    public void setMonth(int month){
12        this.month=month;
13    }
14    public void setYear(int year){
15        this.year=year;
16    }
17    public int getDate(){
18        return date;
19    }
20    public int getMonth(){
21        return month;
22    }
23    public int getYear(){
24        return year;
25    }
26
27    public String toString() {
28        return date+"/"+month+"/"+year;
29    }
30 }
31
```

Lab Manual



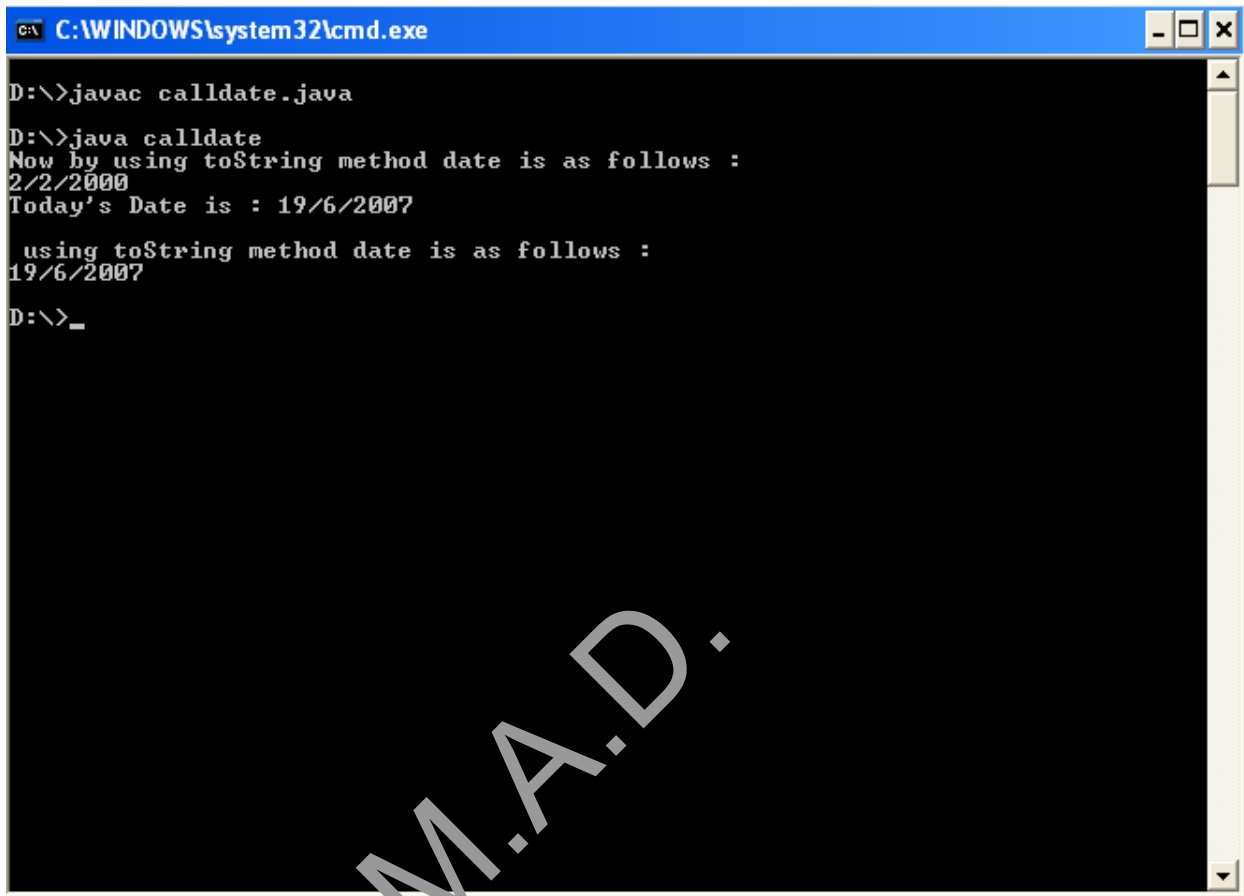
```
60
61 class calldate
62 {
63     public static void main(String[] args)
64     {
65         date d=new date();
66         d.initDate(2,2,2000);
67         System.out.println("Now by using toString method date is as follows : ");
68         System.out.println(d);
69         d.setDate(19);
70         d.setMonth(06);
71         d.setYear(2007);
72
73         System.out.println("Today's Date is : "+d.getDate()+"/"+d.getMonth()+"/"+d.getYear());
74
75         System.out.println();
76         System.out.println("using toString method date is as follows : ");
77         System.out.println(d);
78     }
79 }
80
```

Save as calldate.java

Compile and run calldate.java

Note: When calldate.java is compiled , date class also gets compiled and date.class and calldate.class files get generated

Lab Manual

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The command prompt shows the following text:

```
D:\>javac calldate.java
D:\>java calldate
Now by using toString method date is as follows :
2/2/2000
Today's Date is : 19/6/2007

    using toString method date is as follows :
19/6/2007
D:\>_
```

A large, semi-transparent watermark "M.A.D." is diagonally across the center of the command prompt window.**Assignments To Solve :**

1. WAP to check whether a person is eligible for voting
2. Create a class Book which describes its Book_title and Book_price. Use getter and setter methods to get & set the Books description. Implement createBook and showBookmethods to create and display Book Information. Write a separate class BookInfo to access the Book class
3. Create Model class **Book** with getters and setters, and practice object creation over it
4. How to find leaders element in an array
 1. An element is said to be a leader if all the elements on it's right side are smaller than it. Rightmost element is always a leader. For example, if {14, 9, 11, 7, 8, 5, 3} is the given array then {14, 11, 8, 5, 3} are the leaders in this array (*Bitwise*)
5. Calculate the sum of adjacent numbers of an array, and print the minimum sum (*Vyom labs*)
6. Java Program to find second largest element from an array. (*Vyom labs*)

Lab Manual

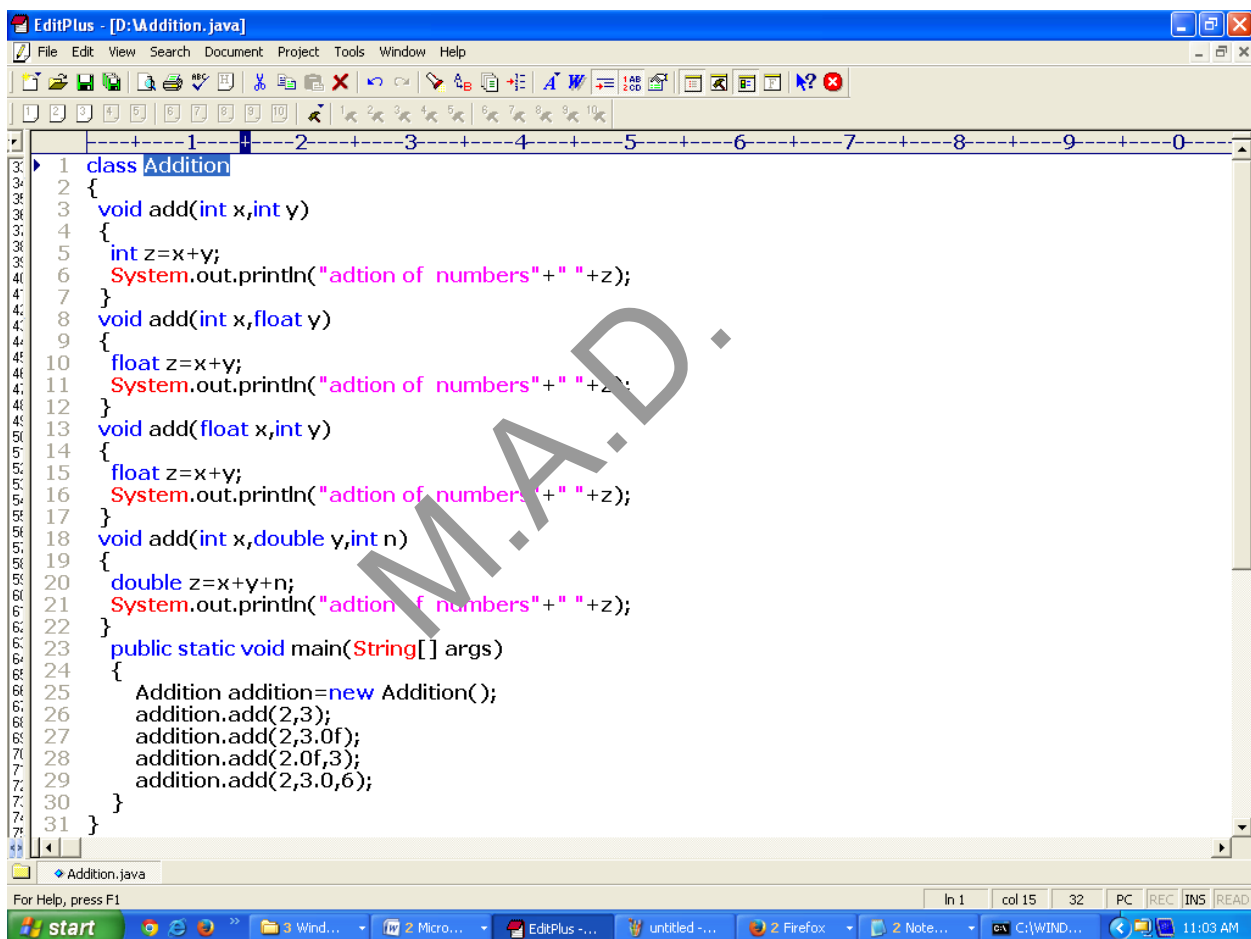
Lab 3

Introducing Constructor

OverLoading Constructors

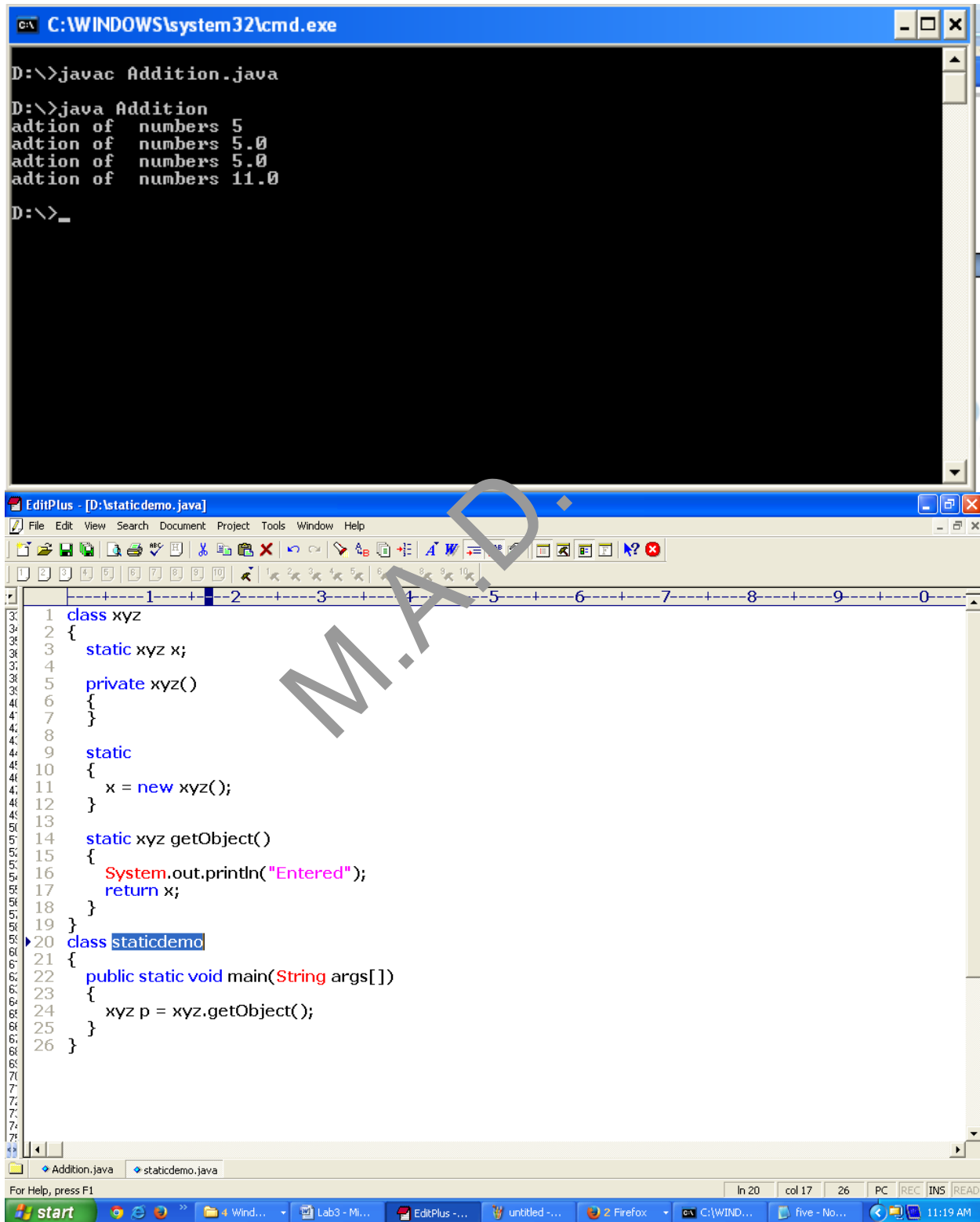
Overloading Methods/varargs/foreach

singleton object



```
1 class Addition
2 {
3     void add(int x,int y)
4     {
5         int z=x+y;
6         System.out.println("adion of numbers"+" "+z);
7     }
8     void add(int x,float y)
9     {
10        float z=x+y;
11        System.out.println("adion of numbers"+" "+z);
12    }
13    void add(float x,int y)
14    {
15        float z=x+y;
16        System.out.println("adion of numbers"+" "+z);
17    }
18    void add(int x,double y,int n)
19    {
20        double z=x+y+n;
21        System.out.println("adion of numbers"+" "+z);
22    }
23    public static void main(String[] args)
24    {
25        Addition addition=new Addition();
26        addition.add(2,3);
27        addition.add(2,3.0f);
28        addition.add(2.0f,3);
29        addition.add(2,3.0,6);
30    }
31 }
```

Lab Manual



The screenshot displays a Windows desktop environment. At the top, a command prompt window titled "C:\WINDOWS\system32\cmd.exe" shows the execution of a Java program. The output of the program is as follows:

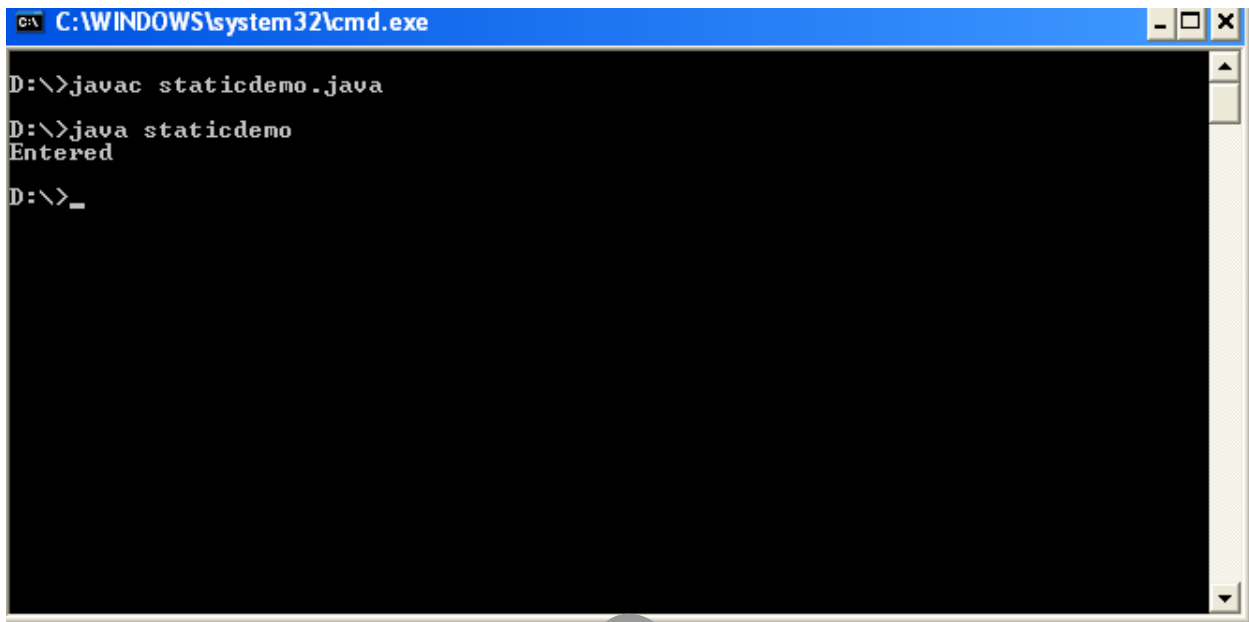
```
D:\>javac Addition.java
D:\>java Addition
addition of numbers 5
addition of numbers 5.0
addition of numbers 5.0
addition of numbers 11.0
D:\>_
```

Below the command prompt, an IDE window titled "EditPlus - [D:\staticdemo.java]" shows the source code of the program. The code is as follows:

```
1 class xyz
2 {
3     static xyz x;
4     private xyz()
5     {
6     }
7     static
8     {
9         x = new xyz();
10    }
11    static xyz getObject()
12    {
13        System.out.println("Entered");
14        return x;
15    }
16 }
17 class staticdemo
18 {
19     public static void main(String args[])
20     {
21         xyz p = xyz.getObject();
22     }
23 }
```

The IDE window also shows a toolbar with various editing and development tools. The taskbar at the bottom of the screen displays the Start button, several open applications (including a web browser, file explorer, and IDE), and the system clock showing 11:19 AM on 11/20/2017.

Lab Manual



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The command prompt shows the following sequence of commands and output:

```
D:\>javac staticdemo.java
D:\>java staticdemo
Entered
D:\>_
```

A large, light gray watermark "M.A.D." is visible diagonally across the lower half of the page.

Assignments To Solve :

1. Write an Account class with default constructor, parameterised constructor and methods toString(), deposit(int amt), withdraw(int amt). withdraw method should take care of insufficient balance. Accept the values from the user..Display the details of various Accounts
2. Write a class Addition2 with add method (overloading) using varargs and enhanced for loop
3. Write a class stack. Use methods push and pop to store and retrieve elements from stack(hint: create an array of int to store the elements)
(Amdocs)
4. Write a singleton class(Singleton class is is a class which has only one object)
(Amdocs)
5. Consider you are working of Mxyz Company and you need to develop a payroll software

A] Construct a class employee with following member using private access specifier:

- | | |
|-----------------|---------|
| 1. employeeId | integer |
| 2. employeeName | string |
| 3. basicSalary | double |
| 4. hra | double |
| 5. medical | double |
| 6. pf | double |
| 7. pt | double |
| 8. netSalary | double |
| 9. grossSalary | double |

- a. Please use following expressions for calculations.
- b. hra=50% of basic salary
- c. pf=12% of basic salary
- d. pt=Rs.200

1. Write no arguments constructor and parameterized constructor to initialize objects.
2. Write properties for employeeId, employeeName, basic salary, netSalary and grossSalary

Lab Manual

3. Write methods to display the details of an employee and calculate the gross and the net salary.

gross salary = basic salary + hra + medical

net salary = gross salary - (pt + pf)

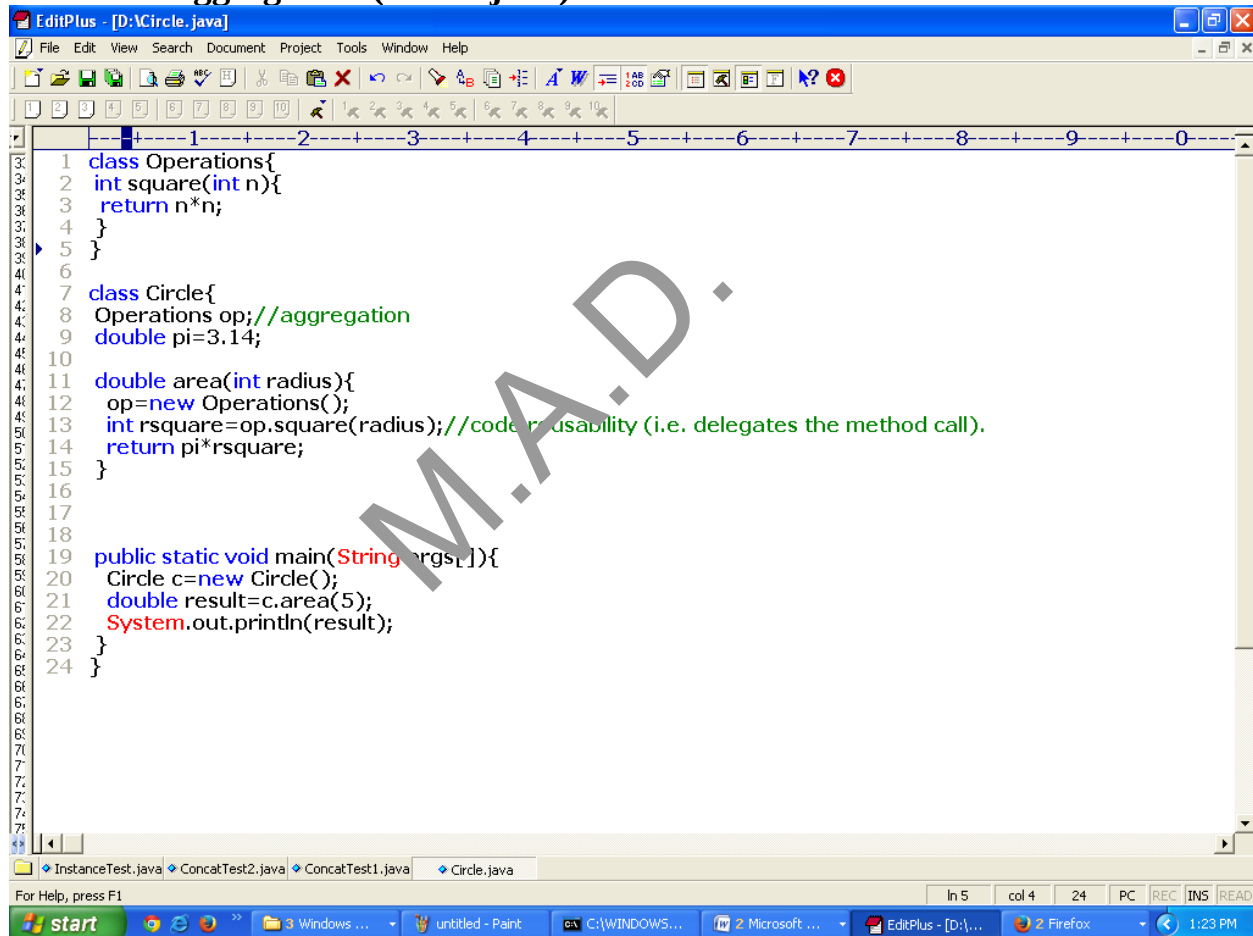
M.A.D.

Lab Manual

Lab 4

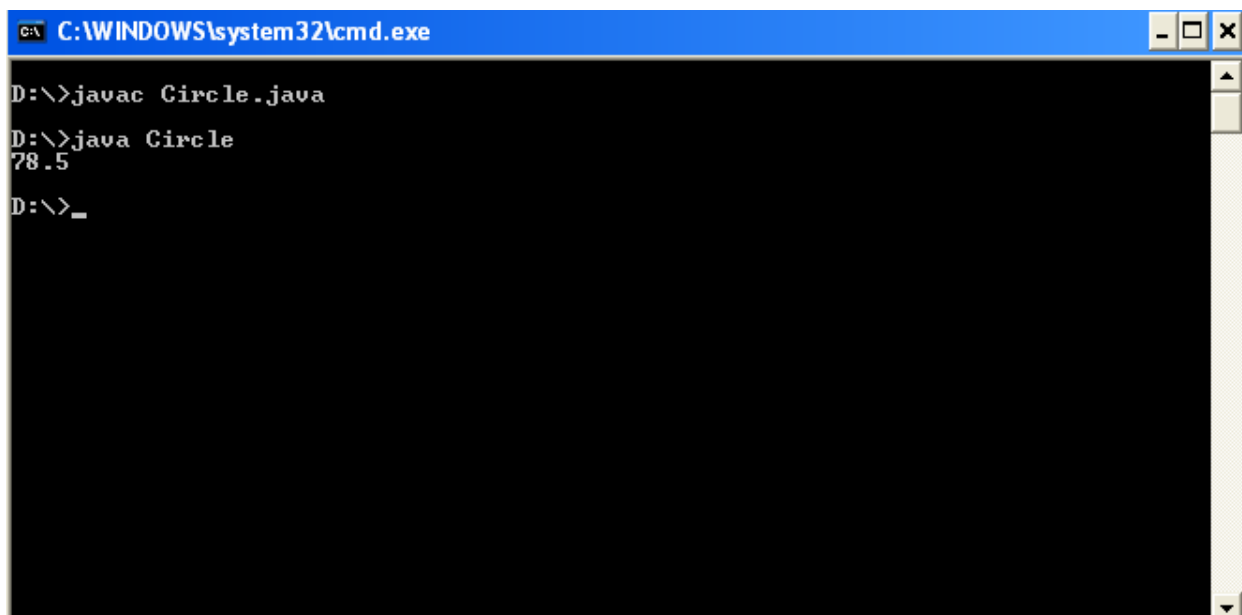
Aggregation/Containment String ,StringBuffer,StringBuilder Math and Wrapper classes Enum

Demo for Aggregation (Circle.java)



```
1 class Operations{
2     int square(int n){
3         return n*n;
4     }
5 }
6
7 class Circle{
8     Operations op;//aggregation
9     double pi=3.14;
10
11     double area(int radius){
12         op=new Operations();
13         int rsquare=op.square(radius);//code reusability (i.e. delegates the method call).
14         return pi*rsquare;
15     }
16
17
18
19     public static void main(String args[]){
20         Circle c=new Circle();
21         double result=c.area(5);
22         System.out.println(result);
23     }
24 }
```

Lab Manual



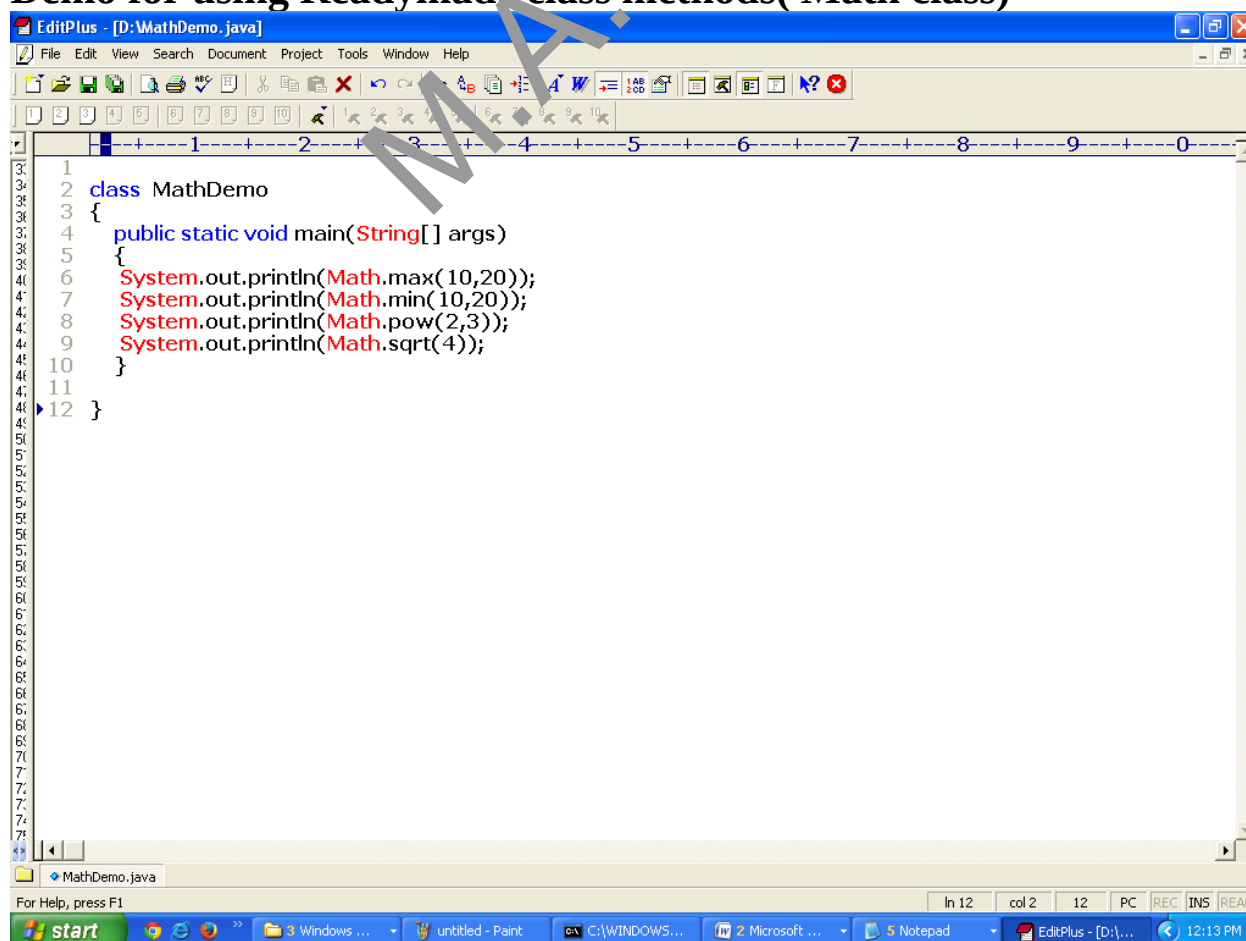
```
C:\WINDOWS\system32\cmd.exe

D:\>javac Circle.java

D:\>java Circle
78.5

D:\>_
```

Demo for using Readymade class methods(Math class)

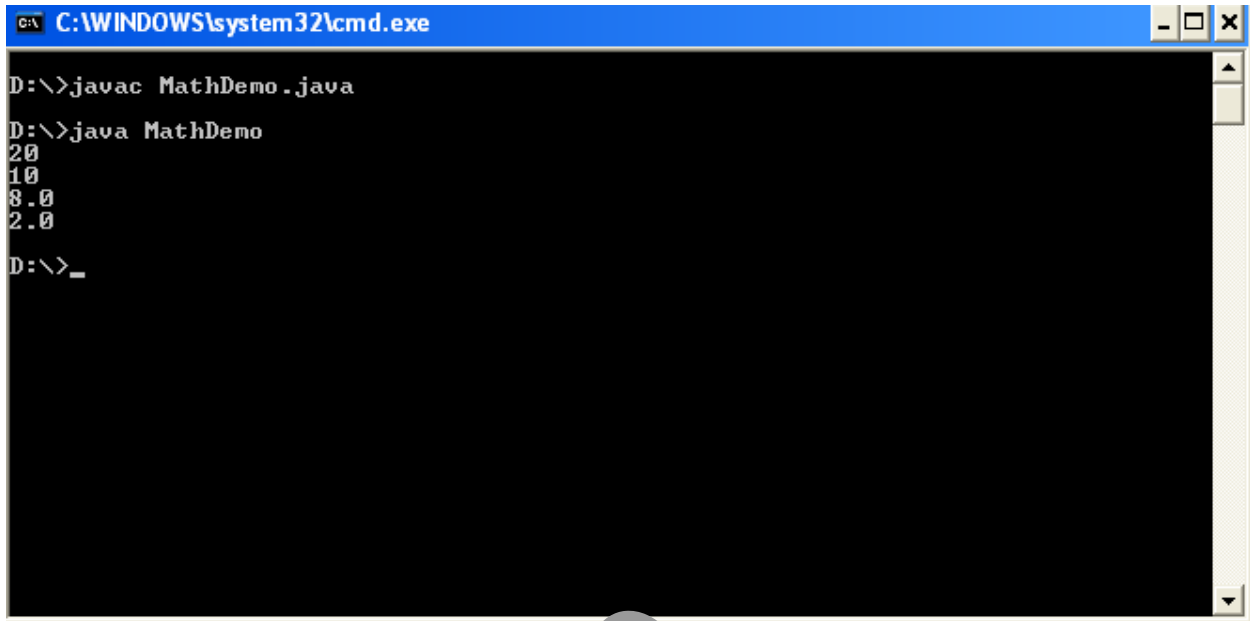


```
EditPlus - [D:\MathDemo.java]
File Edit View Search Document Project Tools Window Help

1
2 class MathDemo
3 {
4     public static void main(String[] args)
5     {
6         System.out.println(Math.max(10,20));
7         System.out.println(Math.min(10,20));
8         System.out.println(Math.pow(2,3));
9         System.out.println(Math.sqrt(4));
10    }
11
12 }
```

Note: Use some more methods of Math class to do math operations

Lab Manual



A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close) on the right. The command prompt shows the following text:

```
D:\>javac MathDemo.java
D:\>java MathDemo
20
10
8.0
2.0
D:\>_
```

The output of the Java program consists of four lines: "20", "10", "8.0", and "2.0". The prompt "D:\>_" indicates that the program has finished execution and the user is back at the command prompt.

M.A.D.

Lab Manual

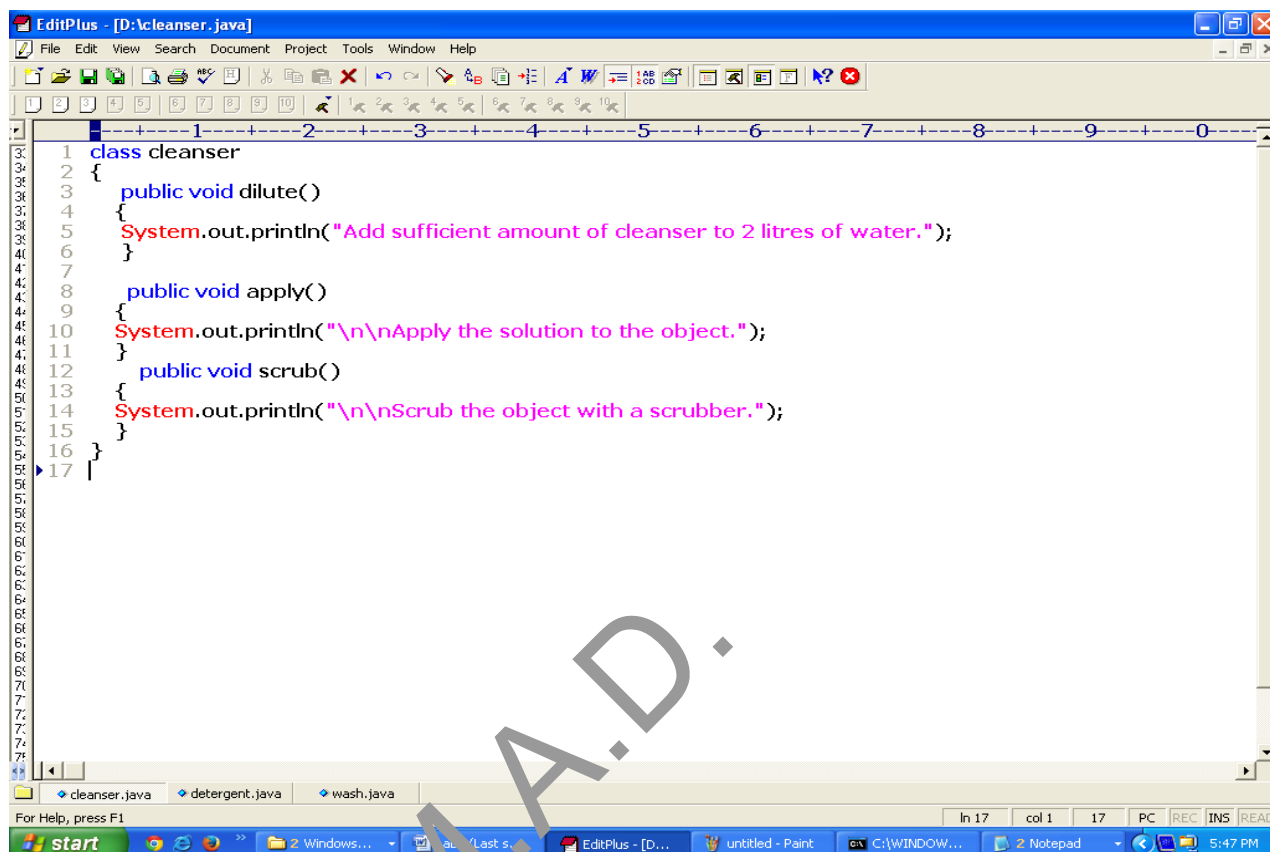
Assignments To Solve :

1. Write an Employee class with id, name and dob(Contained Object.. Containment)with Default and parameterised constructor & toString()..
2. Write program to create array of employee object, and print all employee information
3. Use the String Class Methods like length, hashCode,equals, replace, trim, subString, concatenate , compare , charAt, subString etc.. for a given String(s)
4. practice StringBuffer methods such as length(), capacity(),append(),insert(),delete() etc
5. Write a java program to split string by “blank space” and “-” patterns
6. Create string by using below ways and compare using “==” and equals() method:
 - a. String s1=”abc”;//literal way
 - b. String s2=”abc”;
 - c. String s3=new String(“abc”); //using new keyword
 - d. String s4=new String(‘abc’);
7. consider the given series:1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, ...
 This series is a mixture of 2 series – all the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order. Now write a program to find the Nth term in this series. **(TCS)**
8. Program to sort first half of an array in ascending and second half of an array in descending
(Wipro)
9. Program to check two given matrices are equivalent **(Cognizent)**

Lab 5

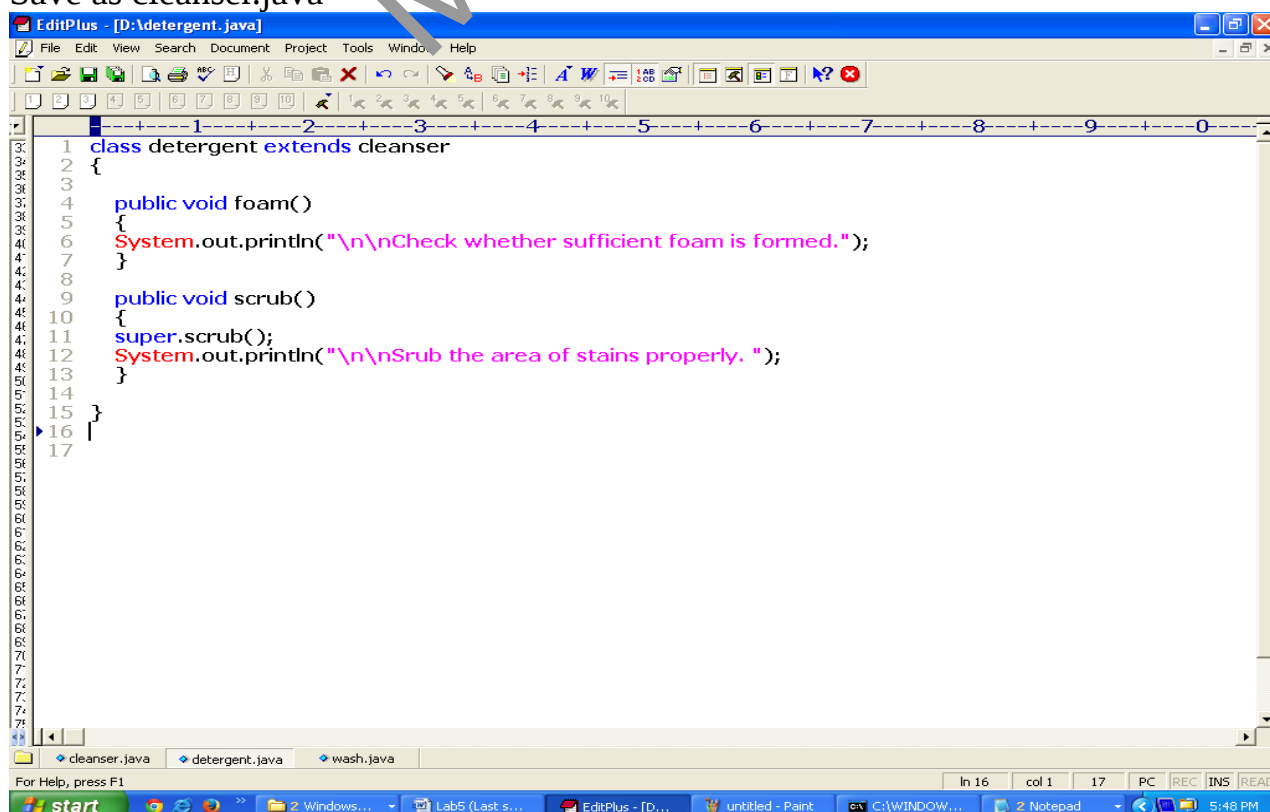
Introducing Inheritance (is a a type of relationship) Demo for Inheritance

Lab Manual



```
1 class cleanser
2 {
3     public void dilute()
4     {
5         System.out.println("Add sufficient amount of cleanser to 2 litres of water.");
6     }
7
8     public void apply()
9     {
10        System.out.println("\n\nApply the solution to the object.");
11    }
12    public void scrub()
13    {
14        System.out.println("\n\nScrub the object with a scrubber.");
15    }
16 }
17
```

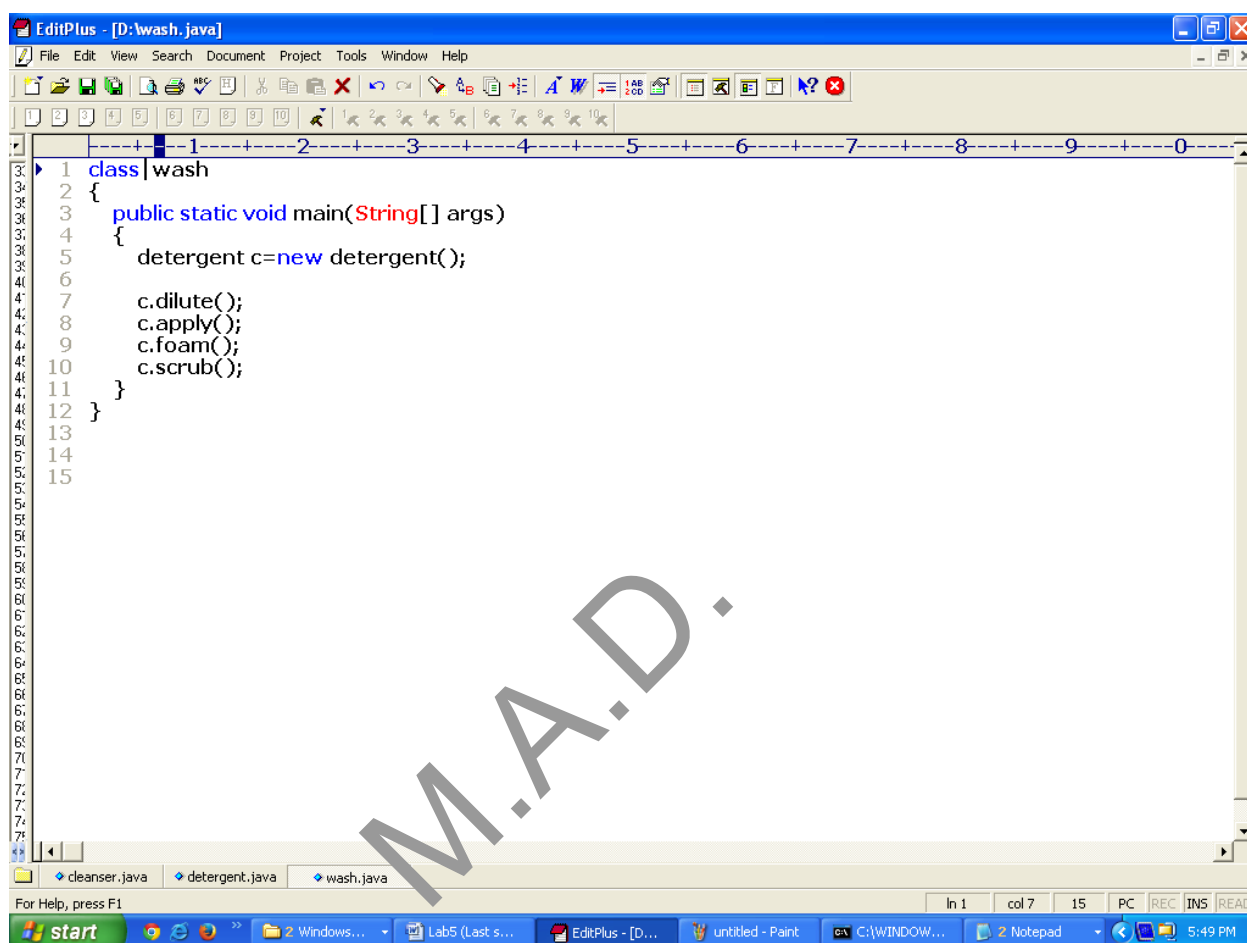
Save as cleanser.java



```
1 class detergent extends cleanser
2 {
3
4     public void foam()
5     {
6         System.out.println("\n\nCheck whether sufficient foam is formed.");
7     }
8
9     public void scrub()
10    {
11        super.scrub();
12        System.out.println("\n\nScrub the area of stains properly. ");
13    }
14 }
15 }
16
17
```

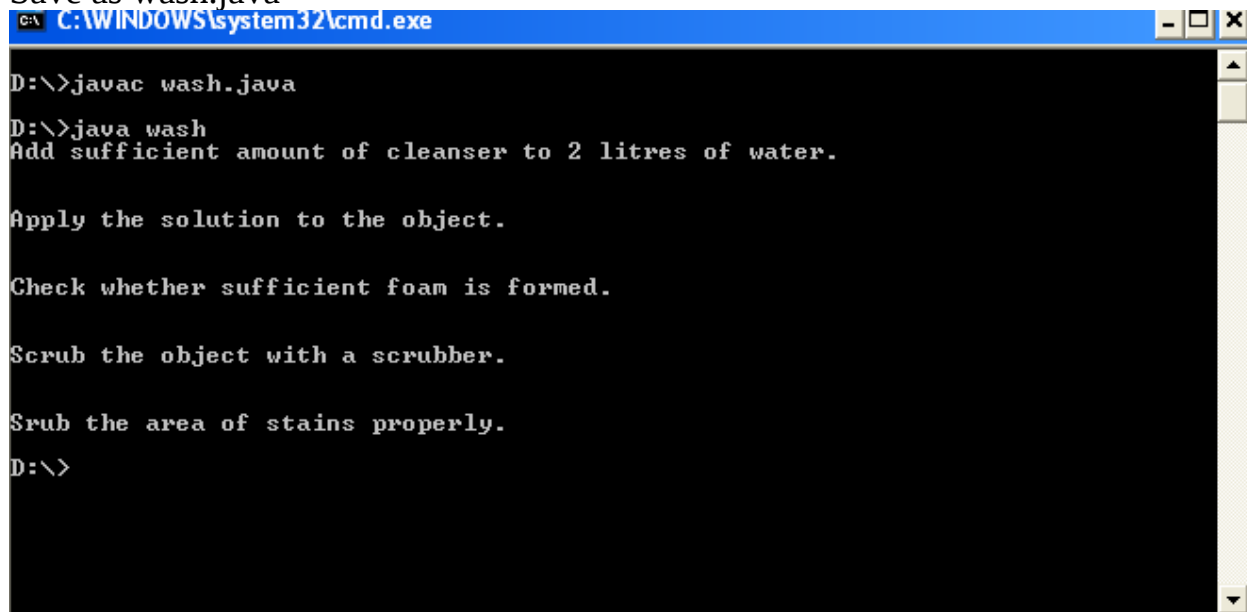
Save as detergent.java

Lab Manual



```
1 class wash
2 {
3     public static void main(String[] args)
4     {
5         detergent c=new detergent();
6
7         c.dilute();
8         c.apply();
9         c.foam();
10        c.scrub();
11    }
12 }
13
14
15
```

Save as wash.java



```
C:\WINDOWS\system32\cmd.exe

D:\>javac wash.java

D:\>java wash
Add sufficient amount of cleanser to 2 litres of water.

Apply the solution to the object.

Check whether sufficient foam is formed.

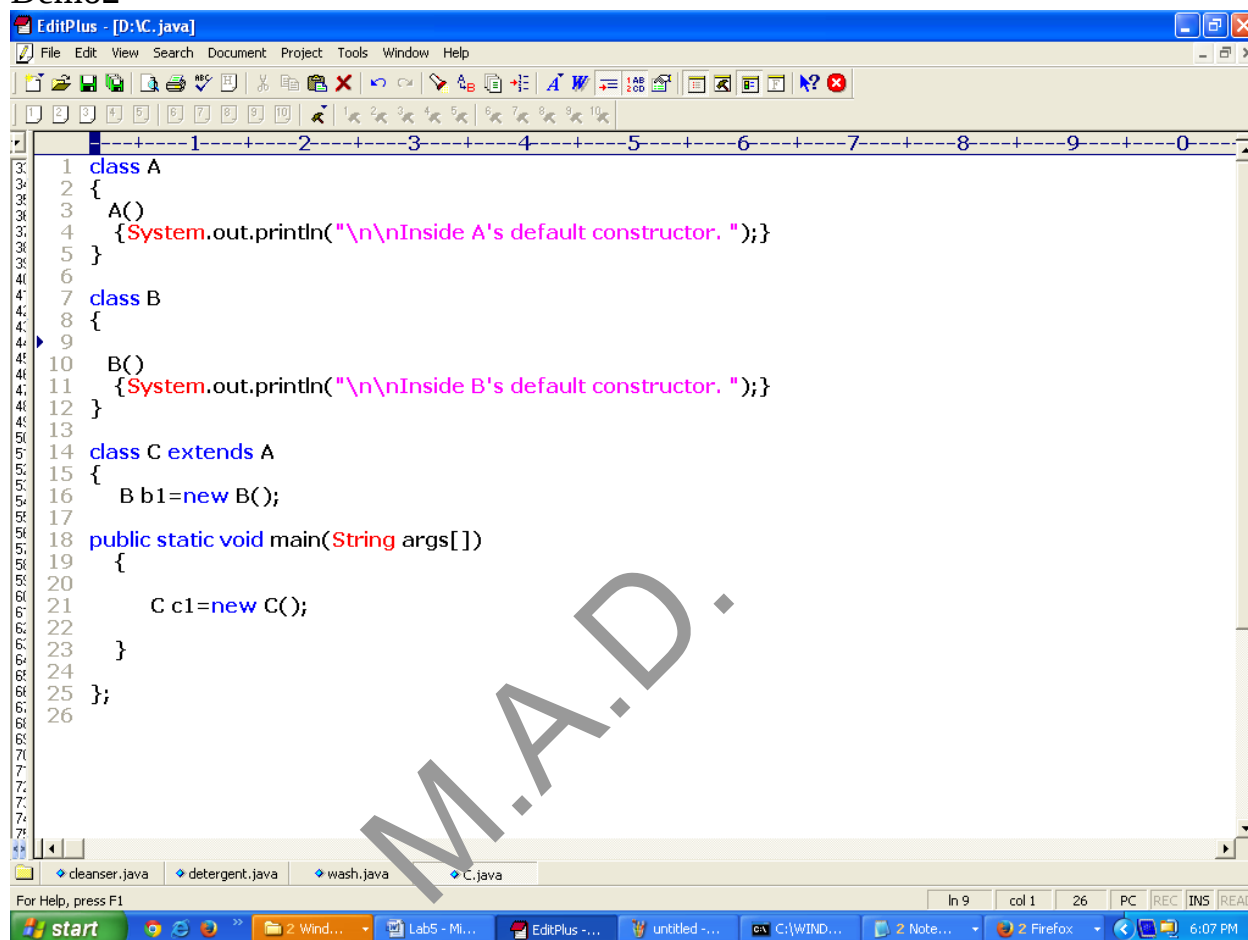
Scrub the object with a scrubber.

Scrub the area of stains properly.

D:\>
```

Lab Manual

Demo2

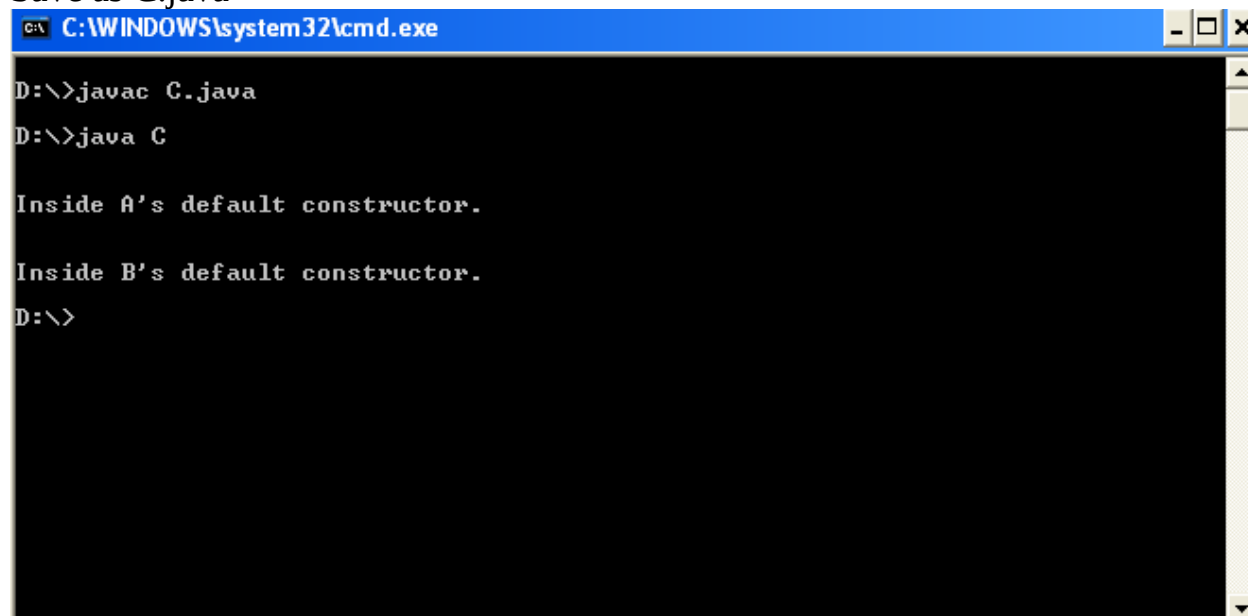


```
1 class A
2 {
3     A()
4     {System.out.println("\n\nInside A's default constructor. ");}
5 }
6
7 class B
8 {
9     B()
10    {System.out.println("\n\nInside B's default constructor. ");}
11 }
12
13 class C extends A
14 {
15     B b1=new B();
16
17     public static void main(String args[])
18     {
19         C c1=new C();
20
21         C c1=new C();
22     }
23 }
24
25 };
26
```

For Help, press F1

start | 2 Wind... | Lab5 - Mi... | EditPlus - ... | untitled - ... | C:\WIND... | 2 Note... | 2 Firefox | 6:07 PM

Save as C.java



```
C:\WINDOWS\system32\cmd.exe

D:\>javac C.java
D:\>java C

Inside A's default constructor.

Inside B's default constructor.
D:\>
```

Assignments To Solve :

1. Write class game{ }

```

class boardgame extends game{ }
class chess extends boardgame{ }
class gamedemo{
public static void main(String[] args) {
chess ch=new chess(); }
}

```

(Note: Write only default constructors in each class with specific information . See the order of constructors invoked when chess object is created)

2. Create class **WageEmployee** extending **Employee** class with attributes as hrs (int)and rate(int) and method computeSalary() to calculate the salary.Print the salary and details of WageEmployee.
(Note: Use the previous date and Employee classes. Accept the values from the user..Default, Parameterised Constructor and toString() to be written in all the classes)

3. Write a java program to remove vowels from a string. (**Wipro**)
4. Str="abceedacbbacc"
write the count of third highest occurring alphabet . (**ADP**)

Lab 6

Introducing Abstract class

Demo for Abstract class

Lab Manual

```

1 abstract class animal {
2     animal()
3     {
4         System.out.println("\nAnimal is the super class");
5     }
6     public abstract void sound();
7 }
8 abstract class herbivores extends animal {
9     herbivores()
10    {
11        System.out.println("\nherbivorous animals depend on plants for their food"); } }
12
13 abstract class carnivores extends animal{
14     carnivores()
15     {
16         System.out.println("\ncarnivorous animals depend on other animals for their food"); } }
17
18 class pig extends herbivores {
19     pig()
20     {System.out.println("\nPig is a herbivorous animal.");}
21     public void sound()
22     {
23         System.out.println("\nPig grunts."); } }
24
25 class tiger extends carnivores {
26     tiger()
27     {System.out.println("\ntiger is a carnivorous animal.");}
28     public void sound()
29     {
30         System.out.println("\n Tiger roars."); } }
31

```

Save as animal.java

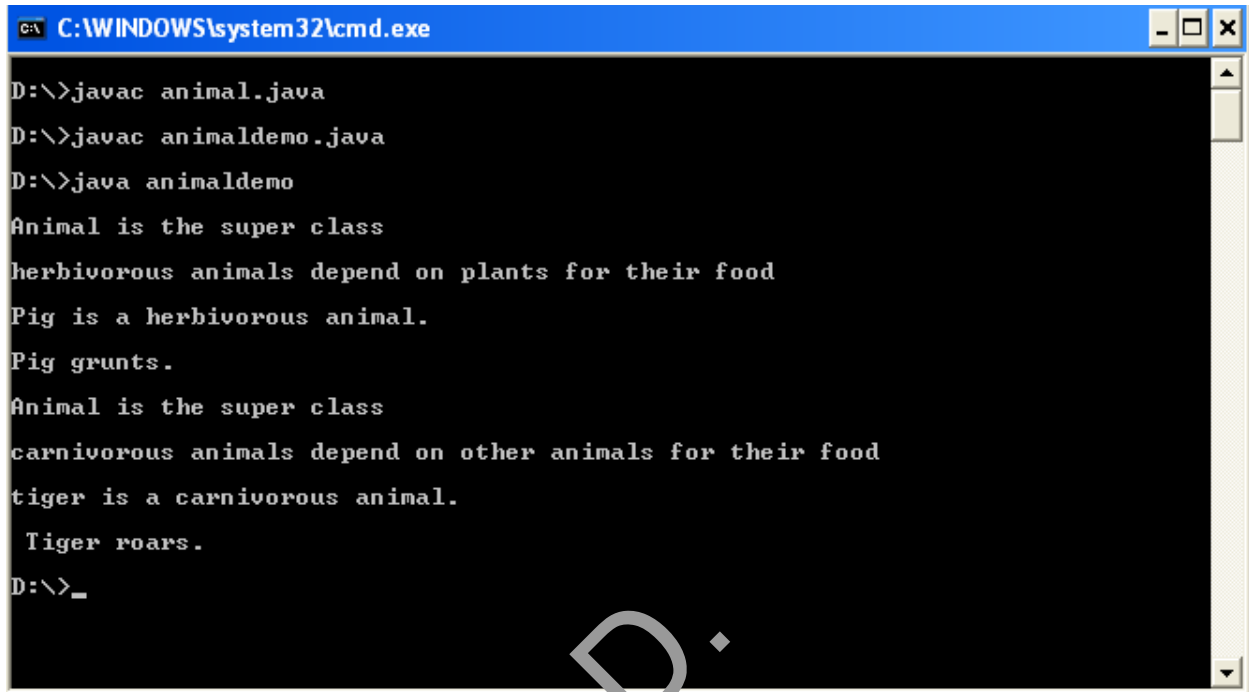
```

1
2 class animaldemo
3 {
4     public static void main(String[] args)
5     {
6         animal a;
7
8         a=new pig();
9         a.sound();
10
11         a=new tiger();
12         a.sound();
13     }
14 }
15

```

Save as animaldemo.java

Lab Manual



```
C:\WINDOWS\system32\cmd.exe

D:\>javac animal.java
D:\>javac animaldemo.java
D:\>java animaldemo
Animal is the super class
herbivorous animals depend on plants for their food
Pig is a herbivorous animal.
Pig grunts.
Animal is the super class
carnivorous animals depend on other animals for their food
tiger is a carnivorous animal.
Tiger roars.
D:\>_
```

A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The command history shows the compilation of 'animal.java' and 'animaldemo.java', followed by the execution of 'java animaldemo'. The output displays class relationships and behaviors for herbivorous and carnivorous animals. A large, diagonal watermark 'M.A.D.' is visible across the lower half of the image.

Assignments To Solve

1. Create an abstract class Instrument which is having the abstract function play.
Create three more sub classes from Instrument which is Piano, Flute, Guitar.
Override the play method inside all three classes printing a message .
“Piano is playing tan tan tan tan ” for Piano class
“Flute is playing toot toot toot toot” for Flute class
“Guitar is playing tin tin tin ” for Guitar class
You must not allow the user to declare an object of Instrument class.
Create an array of 10 Instruments.
Assign different type of instrument to Instrument reference.
Check for the polymorphic behavior of play method.
Use the instanceof operator to print that which object stored at which index of instrument array.
2. Create a class Medicine to represent a drug manufactured by a pharmaceutical company. Provide a function displayLabel() in this class to print Name and address of the company.

Derive Tablet, Syrup and Ointment classes from the Medicine class.
Override the displayLabel() function in each of these classes to print additional information suitable to the type of medicine. For example, in case of tablets, it could be “store in a cool dry place”, in case of ointments it could be “for external use only” etc.
Create a class TestMedicine . Write main function to do the following:
Declare an array of Medicine references of size 10
Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3.
Refer Java API Documentation to find out random generation feature.
Check the polymorphic behavior of the displayLabel() method.
3. Write a java program to display below pattern (**Amdocs**)


```

1
3   2
4   5   6
10  9   8   7
      
```
4. Given a 2D array, print it in spiral form. See the following examples.
(**Cognizent**)
Input:
1 2 3 4

Lab Manual

5 6 7 8

9 10 11 12

13 14 15 16

Output:

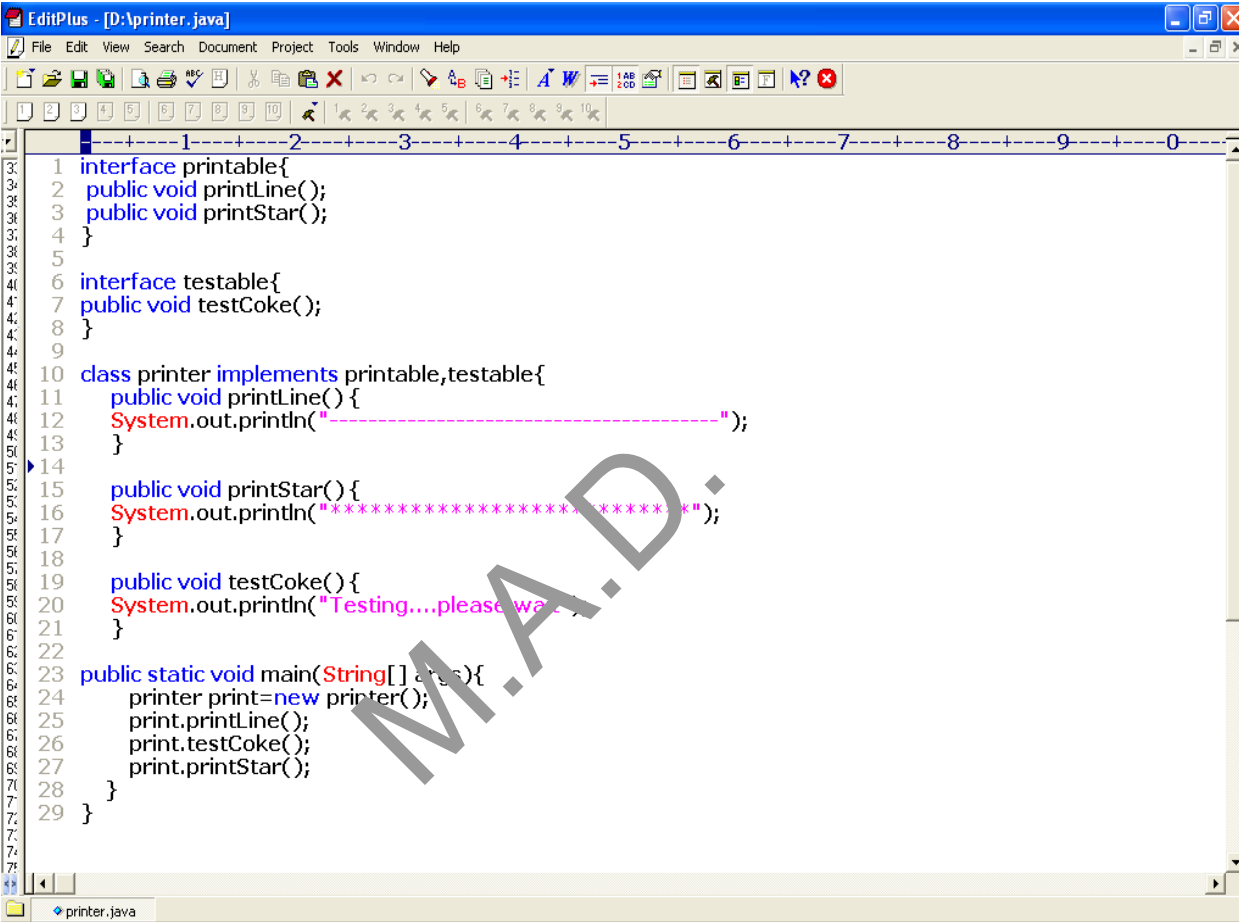
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

M.A.D.

Lab Manual

Lab 7

Introducing Interfaces and Packages



```

1 interface printable{
2     public void printLine();
3     public void printStar();
4 }
5
6 interface testable{
7     public void testCoke();
8 }
9
10 class printer implements printable,testable{
11     public void printLine() {
12         System.out.println("-----");
13     }
14
15     public void printStar() {
16         System.out.println("*****");
17     }
18
19     public void testCoke() {
20         System.out.println("Testing....please wait");
21     }
22
23     public static void main(String[] args){
24         printer print=new printer();
25         print.printLine();
26         print.testCoke();
27         print.printStar();
28     }
29 }

```

printer.java

For Help, press F1

Ln 14 col 1 29 PC REC INS READ

12:11 PM

Save as printer.java

Note: three .class files get created

printable.class

testable.class

printer.class

Lab Manual

```

C:\WINDOWS\system32\cmd.exe

D:\>javac printer.java

D:\>java printer

-----
Testing....please wait
*****
D:\>_

```

Demo2

```

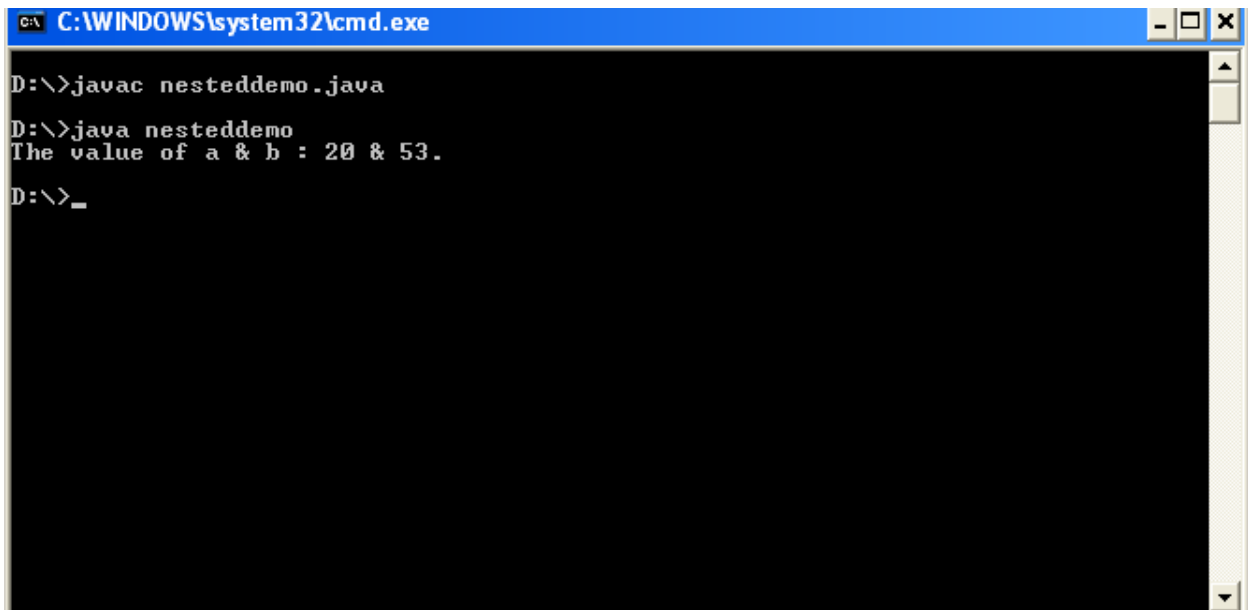
EditPlus - [D:\nesteddemo.java]

1 class nestedinterface {
2     int a;
3
4     nestedinterface(int a) {
5         this.a=a;
6     }
7
8     interface print {
9         void print(String str);
10    }
11
12    class nesteddemo extends nestedinterface implements nestedinterface.print{
13        int b;
14        nesteddemo(int a,int b)
15        {
16            super(a);
17            this.b=b;
18        }
19
20        public void print(String str) {
21            System.out.println(str+a+" & "+b+",");
22        }
23
24        String getdata(){
25            return "The value of a & b : ";
26        }
27
28        public static void main(String args[]){
29            nesteddemo nd=new nesteddemo(20,53);
30            nd.print(nd.getdata()); }
31

```

Save as nesteddemo.java

Lab Manual



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The command prompt shows the following sequence of commands and output:

```
D:\>javac nesteddemo.java
D:\>java nesteddemo
The value of a & b : 20 & 53.
D:\>_
```

A large, light gray watermark "M.A.D." is diagonally oriented across the lower half of the page.

Assignments To Solve

2. Create an interface relational

```

interface relational
{
    void greaterThan();
    void lessThan();
    void greaterThaneq();
    void lessThaneq();
}

```

Write an implementing class implRel which implements relational
 class implRel implements relational

```

{
    int a, b;
    implRel(int a,int b)
    {
        this.a=a;
        this.b=b;
    }

```

```

        - - - -
        - - - -

```

```

}

```

Create a class relationdemo .In main create an object of implRel
 (implementing class)
 and invoke all the methods ...

3. Create an interface

```

interface MyMath
{
    double sqr(double a);
    double cube(double a);
    double cosine(double a);
    double sine(double a);
}

```

Write a class implMath which implements MyMath interface.

Write another class MathDemo ..In main create a reference of MyMath
 pointing to implMath class and invoke the methods..Accept the values
 from the user..

Lab Manual

4. Create a package com.user .

Now create a Greeter class in this package having the following features:

Attributes:

name string //indicates name of the person to be greeted

Member functions:

Greeter(aName)

//constructor to initialize the name of the //person to be greeted by this greeter.

sayHello() //returns a hello message with the name of the //person initialized earlier.

sayGoodBye() //bids goodbye to the person named earlier.

Create another class in the same package called Advisor that has the following features:

Attributes:

message string[5] //contains five advice messages

Member functions:

Advisor() //default constructor to initialize an array of //strings with atleast five advice messages

getAdvice() //randomly selects an advice from the available //list of messages and returns it to the caller of //this method

Outside the package, from your working directory, create a class GreeterTest that constructs Greeter objects for all command-line arguments and prints out the results of calling sayHello().

The program should then display an advice and finally bid goodbye to each of the persons/entities in reverse order of the names entered at the command line.

For e.g.,

```
java GreeterTest Mars Venus
```

then the program should print

Hello, Mars!

Hello, Venus!

Advice: Never say No

Goodbye Venus!

Goodbye Mars!

4. Write a java program to check two strings are anagram or not. **(Culture Machine)**

Example :

Str1="navin";

Str2="vanin"

output :strings are anagram of each other

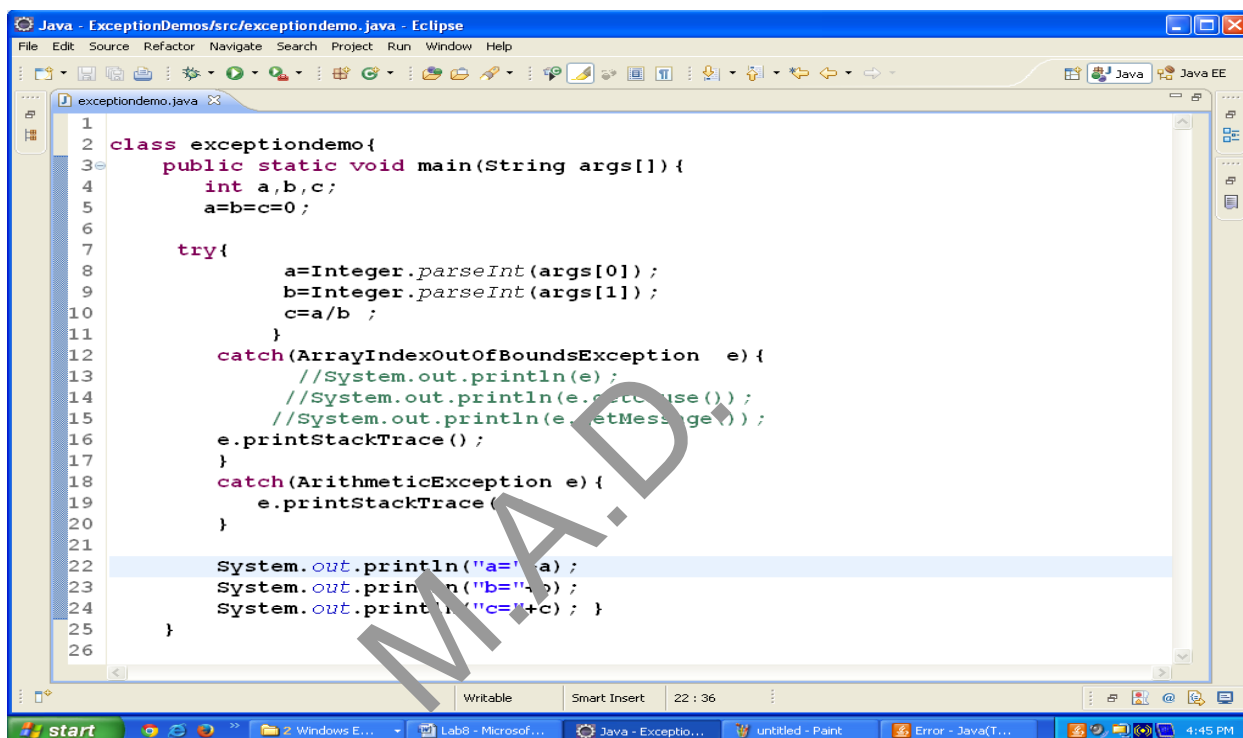
Lab Manual

Lab 8

Exception Handling

Demo1

create exceptiondemo class in Eclipse IDE

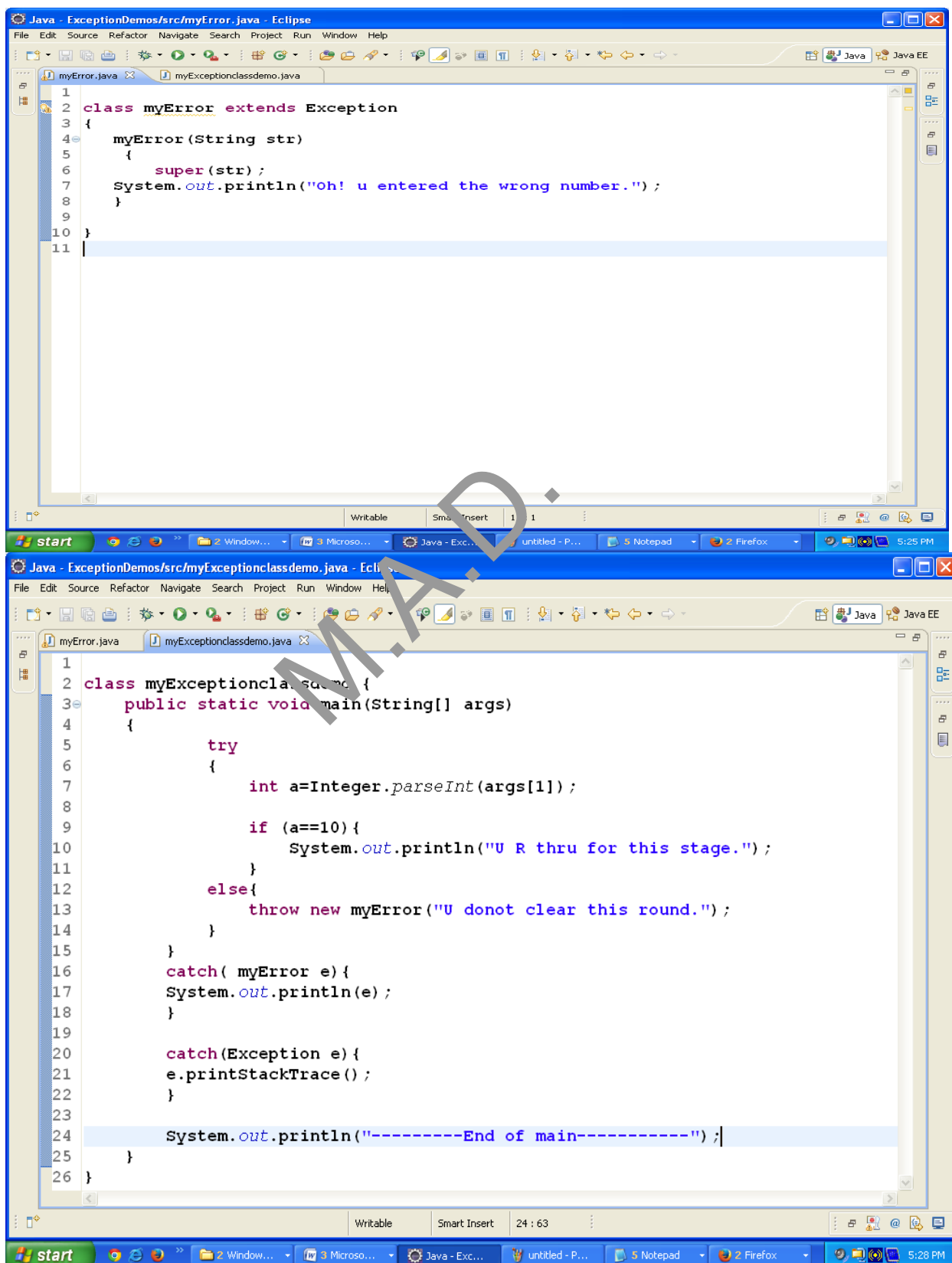
The screenshot shows the Eclipse IDE interface with a Java project named 'ExceptionDemos'. The file 'exceptiondemo.java' is open in the editor. The code defines a class 'exceptiondemo' with a 'main' method. Inside 'main', three integers 'a', 'b', and 'c' are declared and initialized to 0. A 'try' block attempts to parse command-line arguments 'args[0]' and 'args[1]' into integers and calculate 'c = a/b'. Two 'catch' blocks handle 'ArrayIndexOutOfBoundsException' and 'ArithmeticException', both calling 'e.printStackTrace()'. Finally, the values of 'a', 'b', and 'c' are printed. The IDE's status bar at the bottom shows 'Writable', 'Smart Insert', and the cursor position '22 : 36'. The Windows taskbar at the very bottom shows the Start button and several open applications including 'Lab8 - Microsof...', 'Java - Exceptio...', 'Untitled - Paint', and 'Error - Java(T...)'.

Run as -> Java Application

Demo2

create userdefined exception class **myError** and **myExceptiondemoclass**

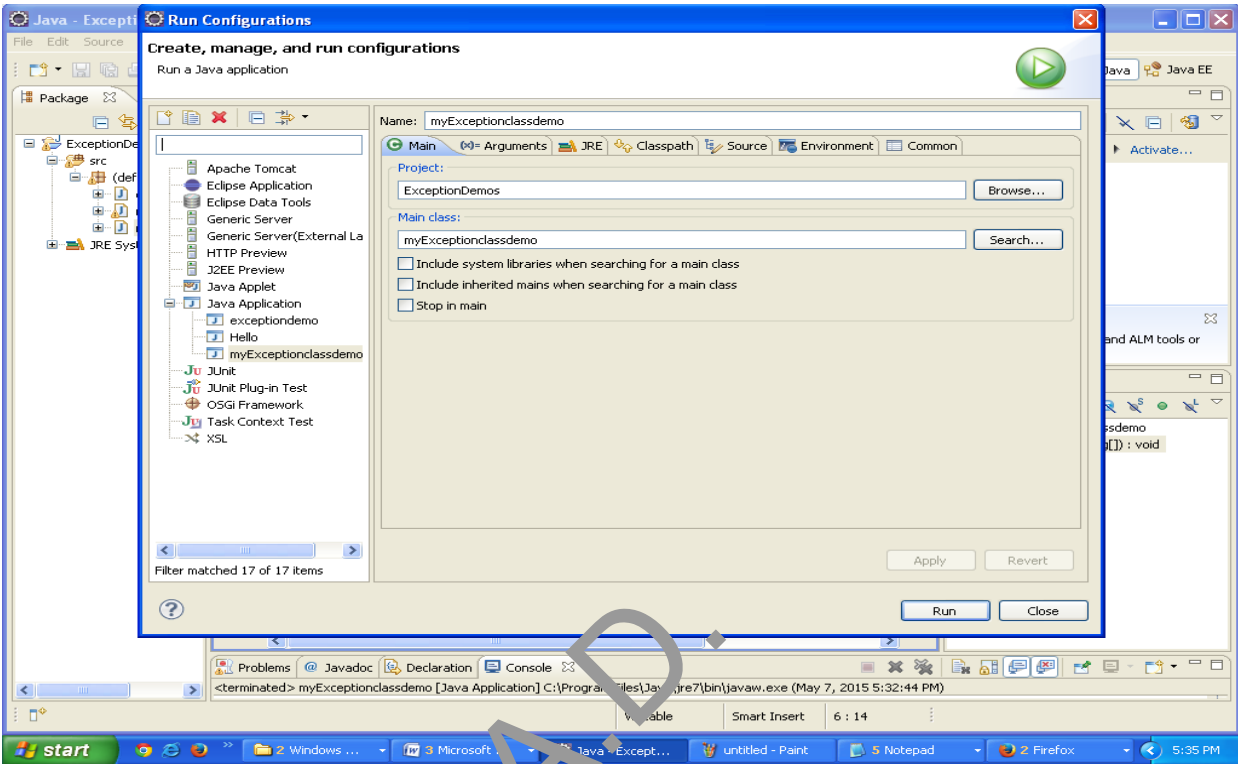
Lab Manual



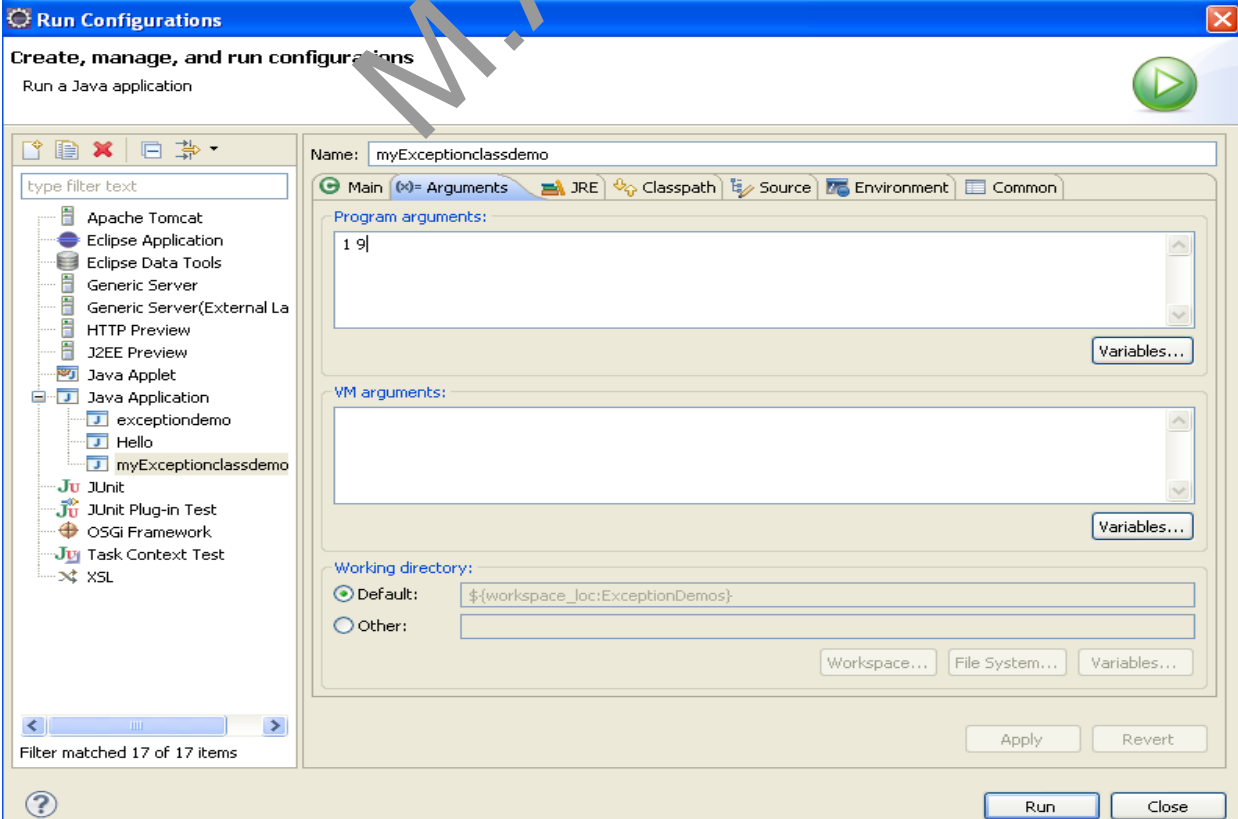
Steps for passing CLA(command Line Arguments in eclipse)

Run as -> Run Configurations...

Lab Manual

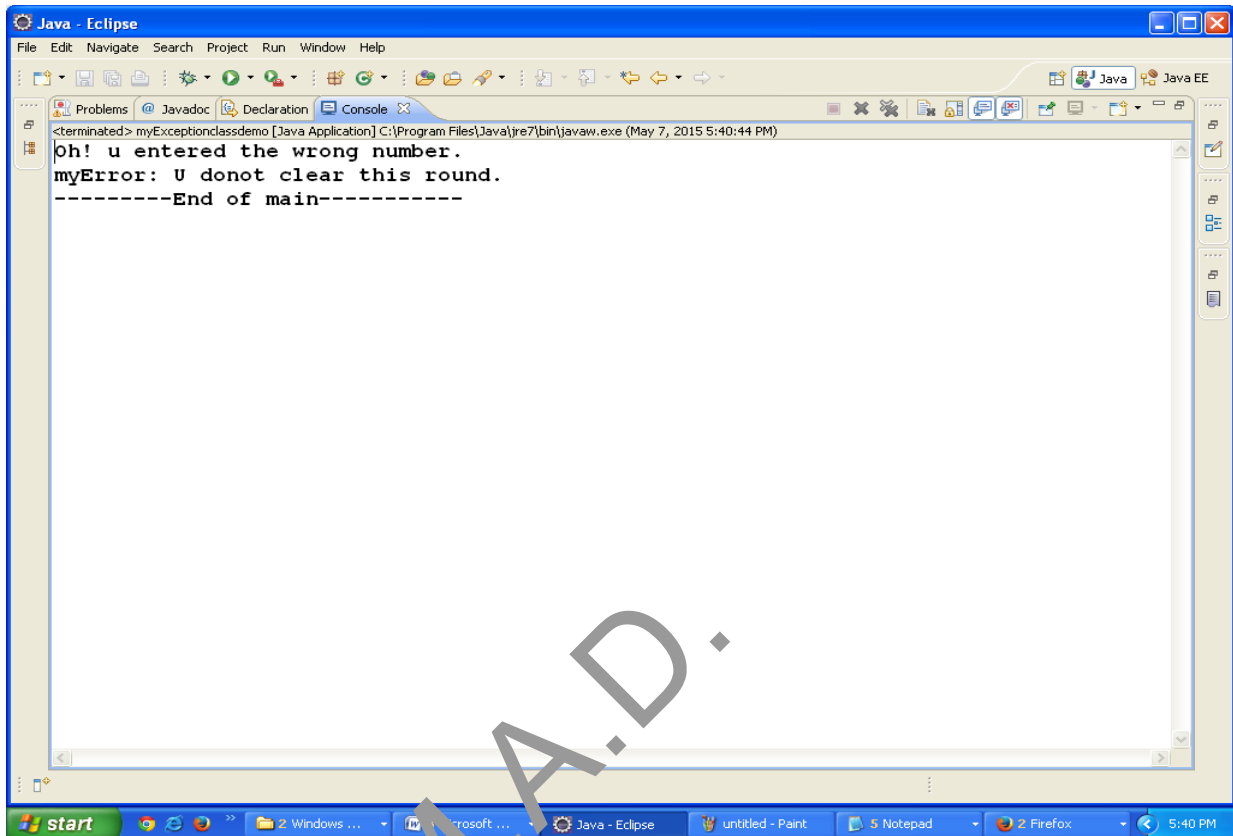


Click on Arguments



Enter Arguments and click on Run

Lab Manual



Assignments To Solve:

1. Write a class **TestException** to test different types of exceptions, such as `NumberFormatException`, `ArrayIndexOutOfBoundsException`, `ArithmeticException`, `NullPointerException`, `ClassNotFoundException` etc

2. Create a class called **CalcAverage** that has the following method:(use throw keyword)

```
public double avgFirstN(int N)
```

This method receives an integer as a parameter and calculates the average of first N natural numbers. If N is not a natural number, throw an exception `IllegalArgumentException` with an appropriate message.

3. Create a class **Number** having the following features:(use throw keyword)

Attributes

int first number

int second number

double result stores result of math operations

performed on a & b Member functions(Methods)

`Number(x, y)` constructor to initialize the values of a and b

`add()` stores the sum of a and b in result

`sub()` stores difference of a and b in result

`mul()` stores product in result

`div()` stores a divided by b in result

Test to see if b is 0 and throw an appropriate exception since division by zero is undefined. Display a menu to the user to perform the above four arithmetic operations.

4. Create a user defined Exception class called **InsufficientFundsException**

(Note: Use the existing **Account** Class (created in lab3))

Handle an exception for **Withdraw(int amt)** in Account class

Withdraw(int amt) should throw **InsufficientFundsException** if the amount to be withdrawn is greater than the balance.

Accept the values from user for creating Account object and amt to withdraw

5. Create userdefined Exception classes **IncorrectAgeException** &

IncorrectNationalityException

Write a class **Voter** with constructors and methods **toString()**,

Lab Manual

check(String name, int age)

The check (...) should check for Nationality and age for voting and throw appropriate Exception

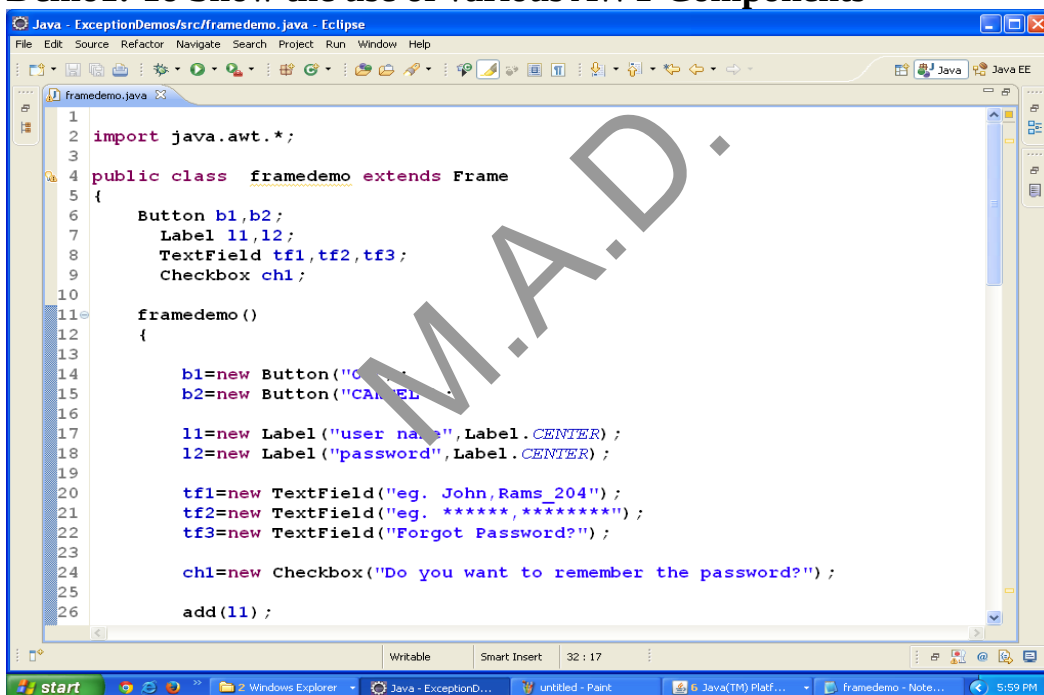
Accept the Name, Nationality & age from user

Lab 9

AWT(Abstract Window ToolKit)

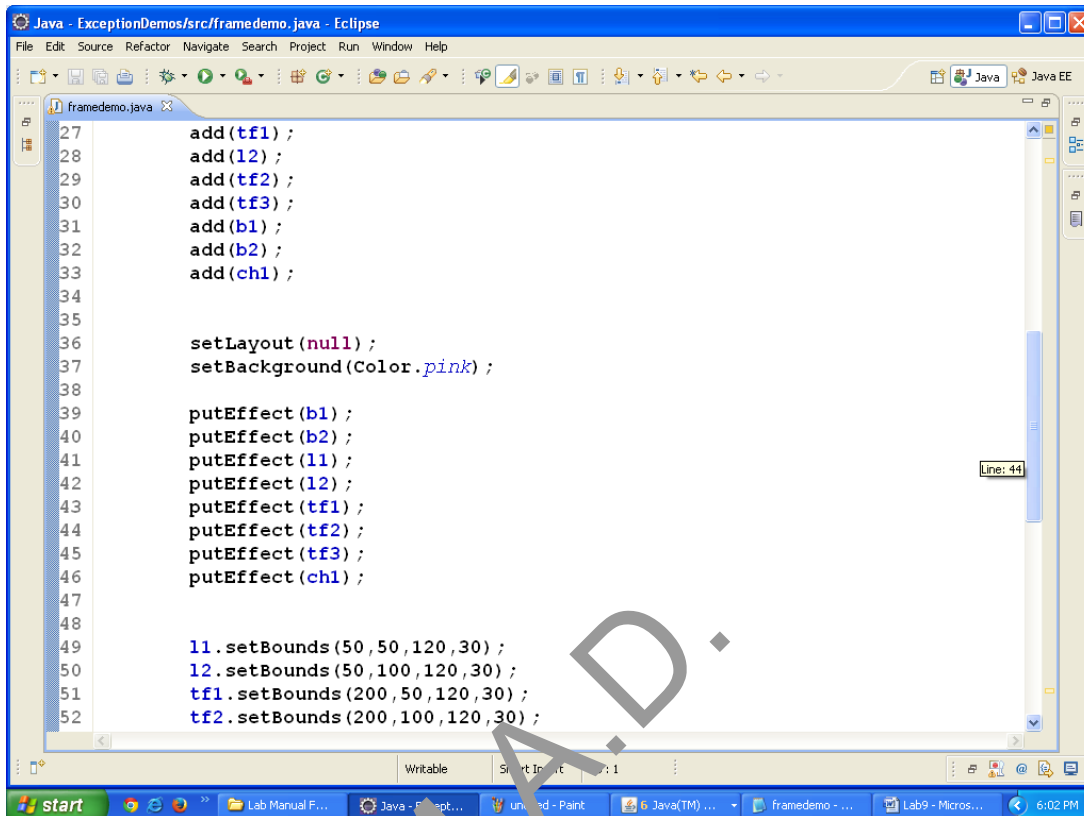
1.Design frames using AWT classes

Demo1: To Show the use of various AWT Components



```
1 import java.awt.*;
2
3
4 public class framedemo extends Frame
5 {
6     Button b1,b2;
7     Label l1,l2;
8     TextField tf1,tf2,tf3;
9     Checkbox chl;
10
11     framedemo ()
12     {
13
14         b1=new Button("OK");
15         b2=new Button("CANCEL");
16
17         l1=new Label ("user name",Label.CENTER);
18         l2=new Label ("password",Label.CENTER);
19
20         tf1=new TextField("eg. John,Rams_204");
21         tf2=new TextField("eg. *****,*****");
22         tf3=new TextField("Forgot Password?");
23
24         chl=new Checkbox("Do you want to remember the password?");
25
26         add(l1);
```

Lab Manual



Lab Manual

```

53      tf3.setBounds(50,150,220,30);
54      b1.setBounds(50,200,50,30);
55      b2.setBounds(120,200,100,30);
56      chl.setBounds(50,250,250,30);
57
58      tf3.setEditable(false);
59
60      chl.setFont(new Font("Times New Roman",Font.PLAIN,12));
61
62      setSize(400,400);
63      setVisible(true);
64  }
65
66  public void putEffect(Component c)
67  {
68      c.setBackground(Color.white);
69      c.setForeground(Color.black);
70      c.setFont(new Font("Arial Narrow",Font.PLAIN,14));
71  }
72
73  public static void main(String[] args)
74  {
75      framedemo frd=new framedemo();
76  }
77 }
78

```

Run as -> Java Application

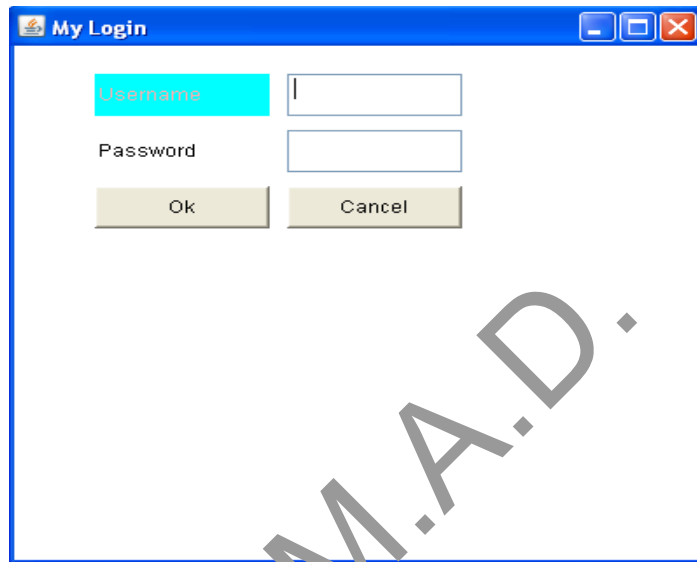
The application window has a pink background and contains the following elements:

- user name**: A text field with the placeholder text "eg. John,Rams_2l".
- password**: A text field with the placeholder text "eg. *****,*****".
- Forgot Password?**: A button.
- OK** and **CANCEL**: Two buttons.
- ☐ **Do you want to remember the password?**: A checkbox with a label.

Lab Manual

Assignments To Solve

1. Generate the Login Page as shown below. (Event for which should be handled in the next lab..)

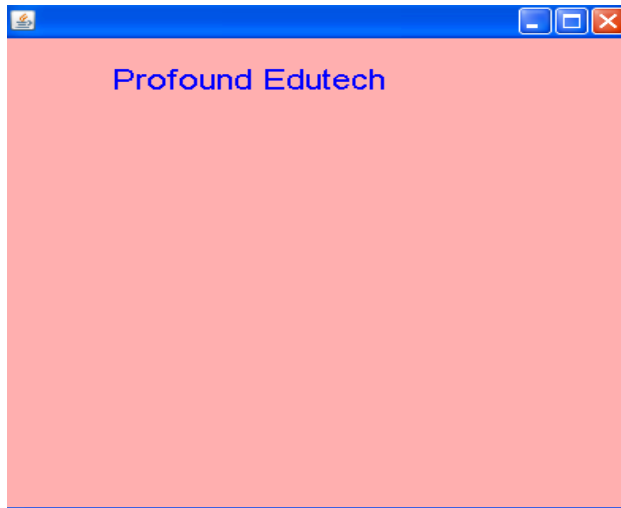


2. Generate the below page.. Draw 3 Scrollbars with backgrounds.(Event for which should be handled in the next lab..)



3. WAP to print your name with the particular Font on the Frame

Lab Manual

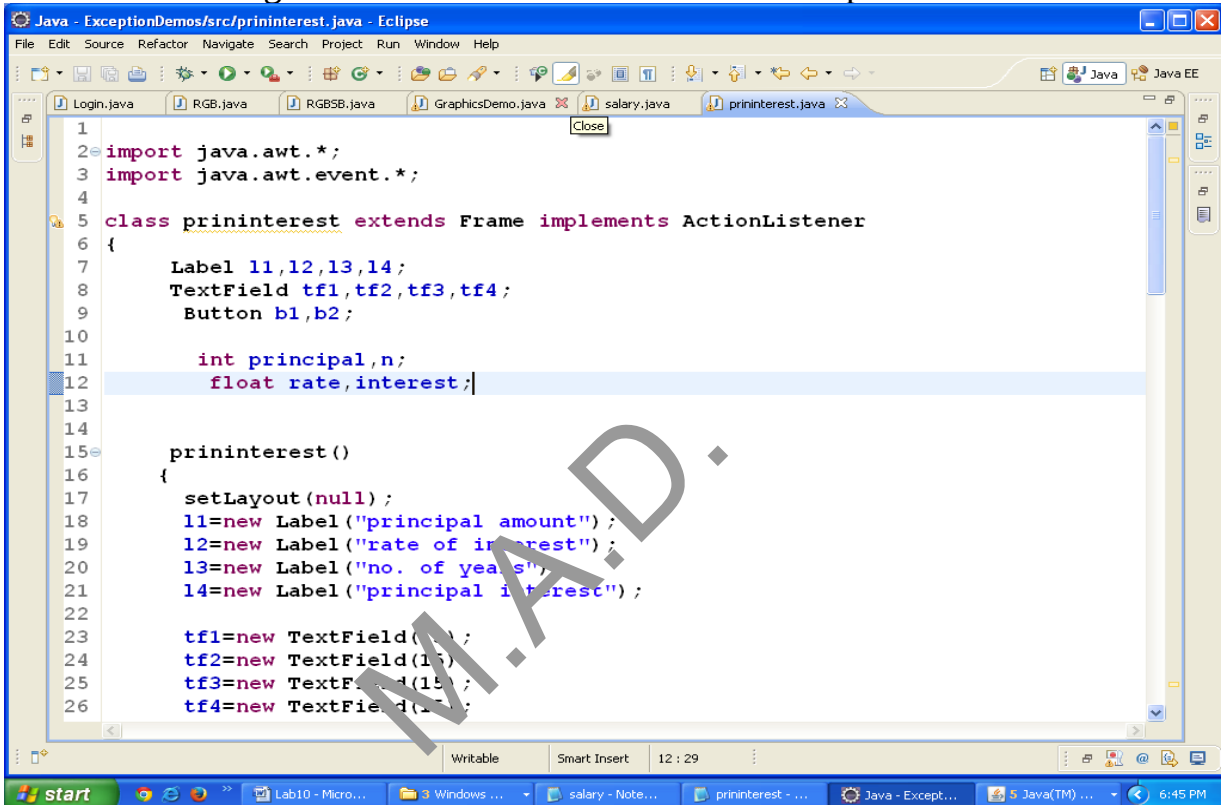


M.A.D.

Lab Manual

Event Handling

Demo 1: Handling an Event for Button to calculate SimpleInterest

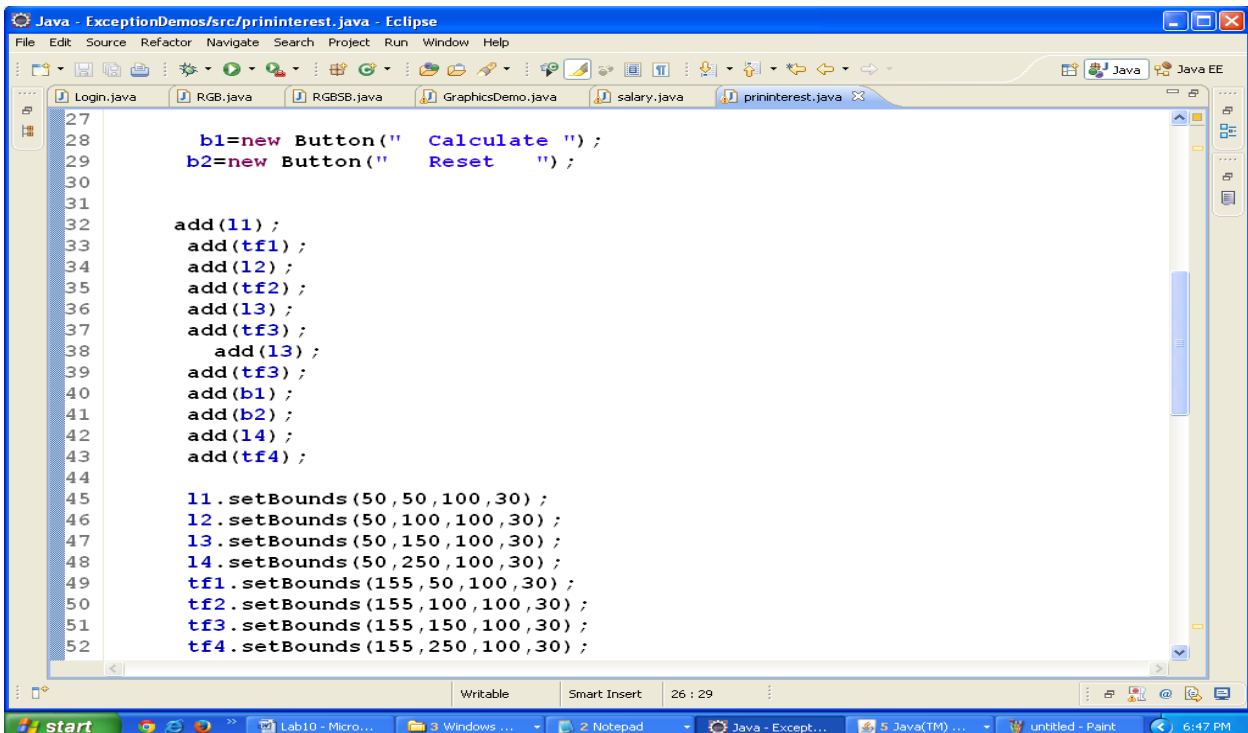


```

1
2 import java.awt.*;
3 import java.awt.event.*;
4
5 class prininterest extends Frame implements ActionListener
6 {
7     Label l1,l2,l3,l4;
8     TextField tf1,tf2,tf3,tf4;
9     Button b1,b2;
10
11     int principal,n;
12     float rate,interest;
13
14
15     prininterest()
16     {
17         setLayout(null);
18         l1=new Label("principal amount");
19         l2=new Label("rate of interest");
20         l3=new Label("no. of years");
21         l4=new Label("principal interest");
22
23         tf1=new TextField(10);
24         tf2=new TextField(10);
25         tf3=new TextField(10);
26         tf4=new TextField(10);

```

Continued...



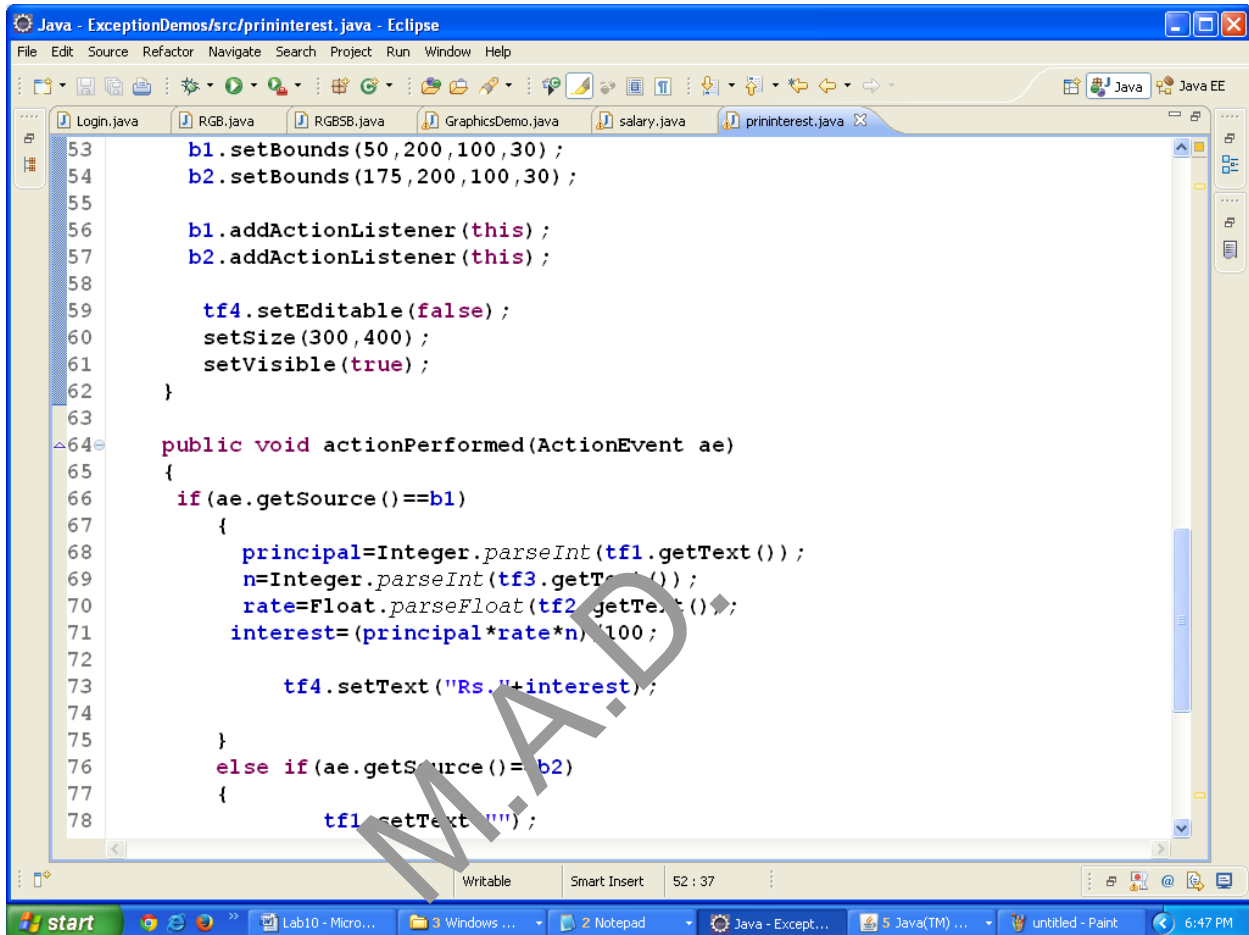
```

27
28     b1=new Button(" Calculate ");
29     b2=new Button(" Reset ");
30
31
32     add(l1);
33     add(tf1);
34     add(l2);
35     add(tf2);
36     add(l3);
37     add(tf3);
38     add(l3);
39     add(tf3);
40     add(b1);
41     add(b2);
42     add(l4);
43     add(tf4);
44
45     l1.setBounds(50,50,100,30);
46     l2.setBounds(50,100,100,30);
47     l3.setBounds(50,150,100,30);
48     l4.setBounds(50,250,100,30);
49     tf1.setBounds(155,50,100,30);
50     tf2.setBounds(155,100,100,30);
51     tf3.setBounds(155,150,100,30);
52     tf4.setBounds(155,250,100,30);

```

Lab Manual

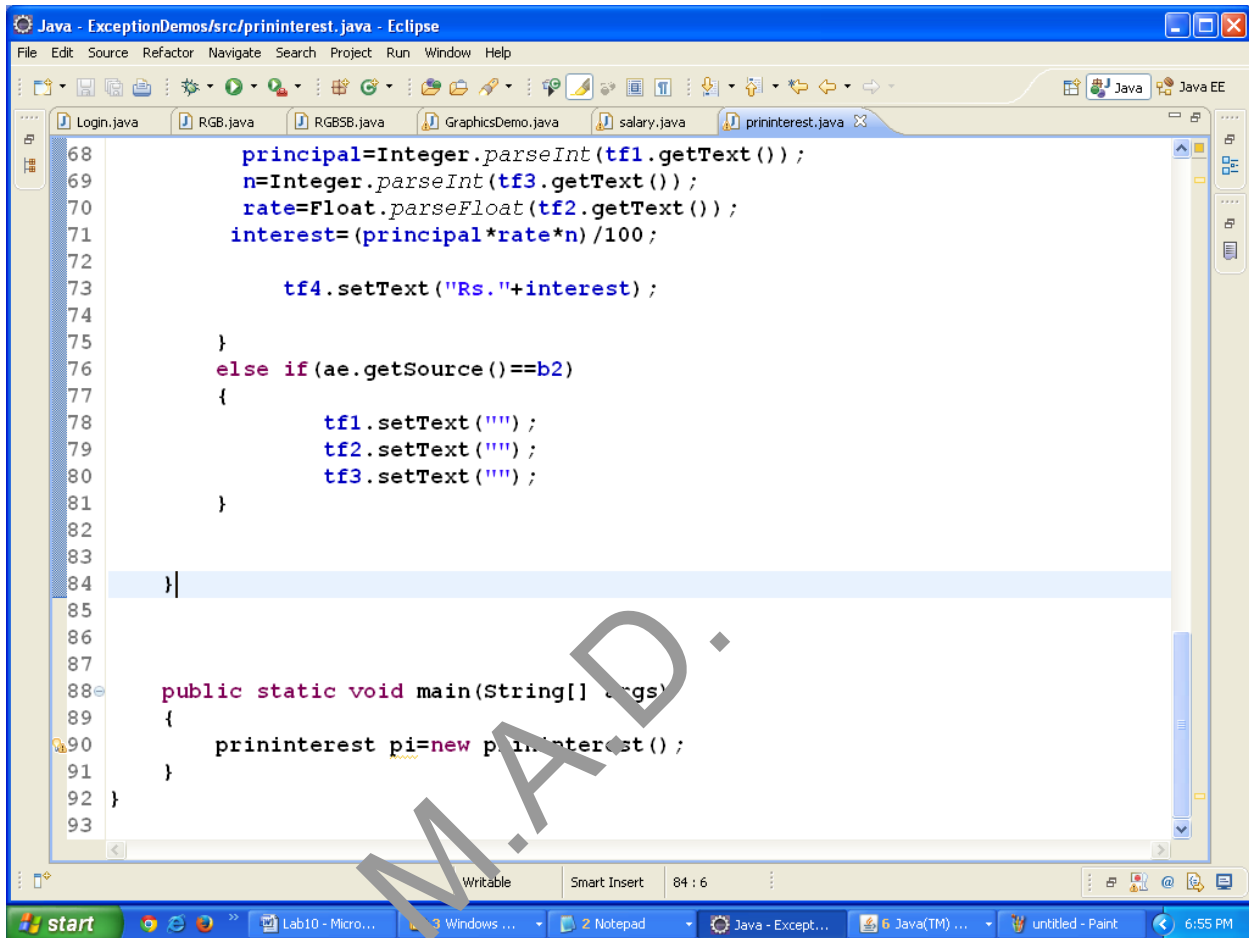
Continued...



```
53      b1.setBounds(50,200,100,30);
54      b2.setBounds(175,200,100,30);
55
56      b1.addActionListener(this);
57      b2.addActionListener(this);
58
59      tf4.setEditable(false);
60      setSize(300,400);
61      setVisible(true);
62  }
63
64  public void actionPerformed(ActionEvent ae)
65  {
66      if(ae.getSource()==b1)
67      {
68          principal=Integer.parseInt(tf1.getText());
69          n=Integer.parseInt(tf3.getText());
70          rate=Float.parseFloat(tf2.getText());
71          interest=(principal*rate*n)/100;
72
73          tf4.setText("Rs."+interest);
74
75      }
76      else if(ae.getSource()==b2)
77      {
78          tf1.setText("");
79      }
79  }
```

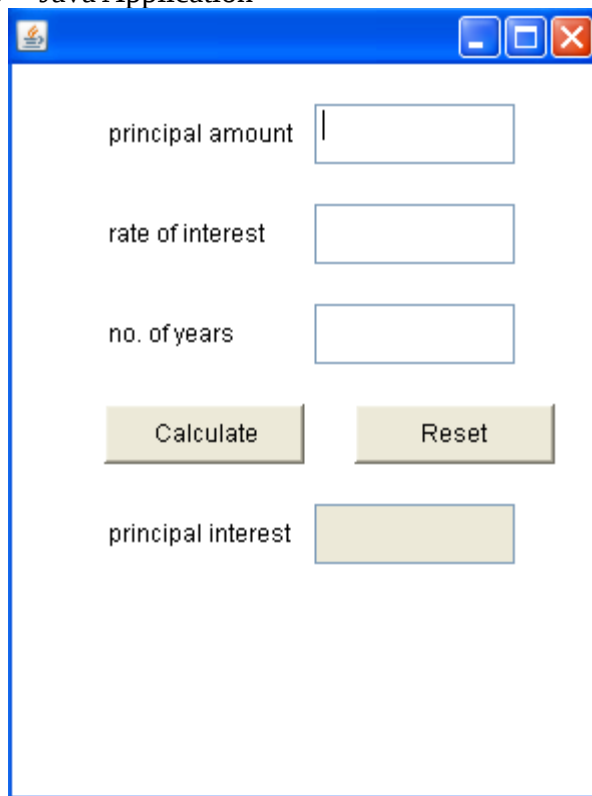
Continued...

Lab Manual



```
68      principal=Integer.parseInt(tf1.getText());
69      n=Integer.parseInt(tf3.getText());
70      rate=Float.parseFloat(tf2.getText());
71      interest=(principal*rate*n)/100;
72
73      tf4.setText("Rs."+interest);
74
75  }
76  else if(ae.getSource()==b2)
77  {
78      tf1.setText("");
79      tf2.setText("");
80      tf3.setText("");
81  }
82
83
84  }
85
86
87
88  public static void main(String[] args)
89  {
90      prininterest pi=new prininterest();
91  }
92  }
93
```

Run as-> Java Application



principal amount

rate of interest

no. of years

principal interest

Lab Manual

Enter the values and click on Calculate Button(Event Handling)

principal amount 5000

rate of interest 9.5

no. of years 5

Calculate Reset

principal interest Rs.2375.0

Assignments To Solve

1. Validate Username and Password for LoginPage created in the previous lab
Print Access Granted on the Frame for valid data
Print Access Denied on the Frame for Invalid data
Handle this event for the click of **OK** Button
Close the Frame at the click of **Cancel** Button
2. Write an event handling code to print your name on the frame at the click of Mouse Button.(hint : Mouse Event)
3. Handle an event for the Scrollbars created in the previous lab.
Set the background color of the frame with the Scroll values of the 3 Scrollbars.
4. Handle an event for the TextFields shown below . Show the Focus gained

Lab Manual

for the TextField when TextField is selected using Tab..(HINT: Use FocusListener)

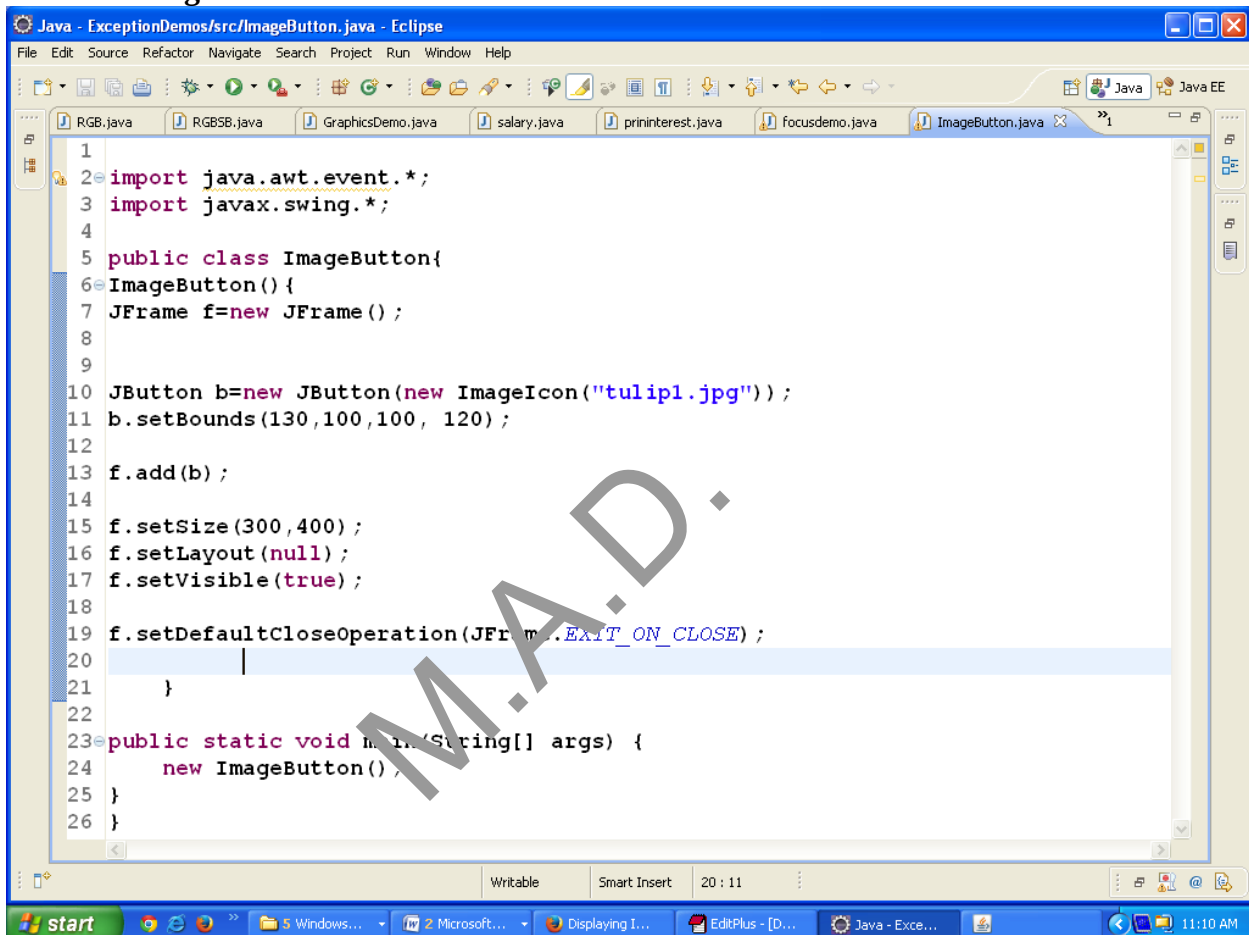


Lab Manual

Swing Components

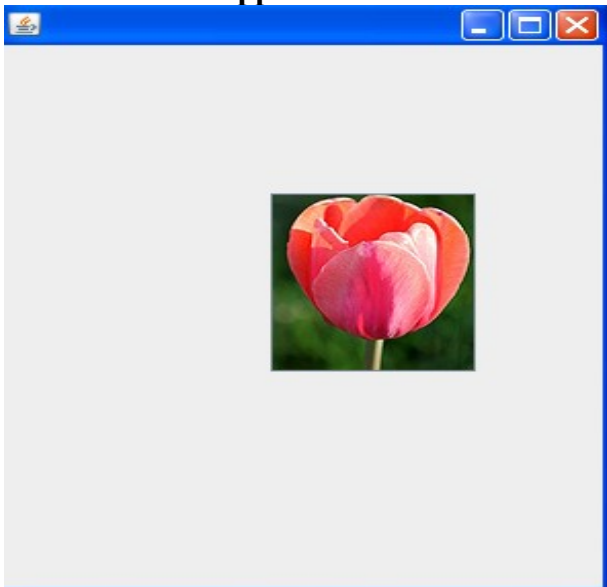
MenuBar

Demo 1: ImageButton



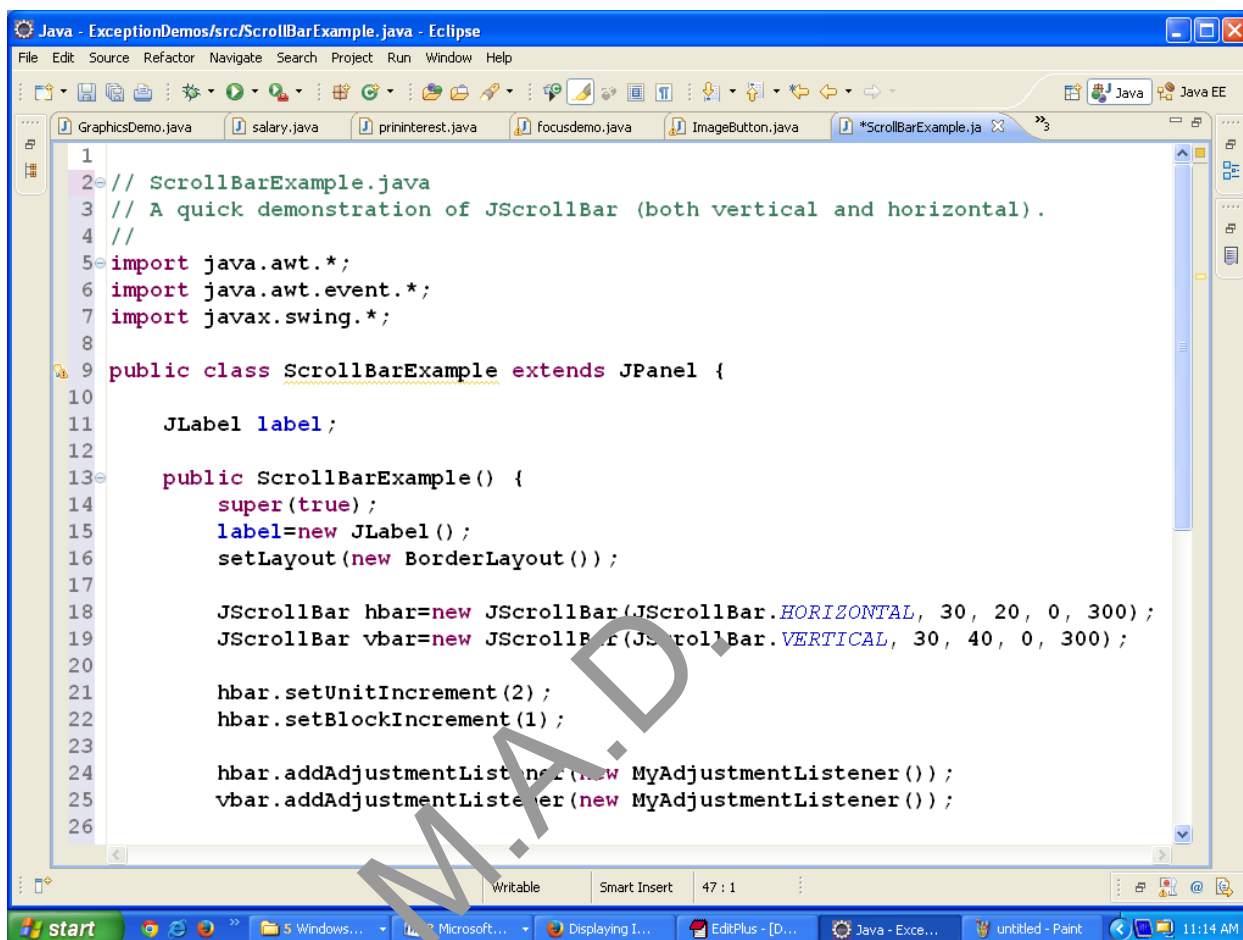
```
1
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class ImageButton{
6 ImageButton() {
7 JFrame f=new JFrame();
8
9
10 JButton b=new JButton(new ImageIcon("tulip1.jpg"));
11 b.setBounds(130,100,100, 120);
12
13 f.add(b);
14
15 f.setSize(300,400);
16 f.setLayout(null);
17 f.setVisible(true);
18
19 f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20
21 }
22
23 public static void main(String[] args) {
24     new ImageButton();
25 }
26 }
```

Run as -> Java Application



Demo 2: ScrollBarDemo

Lab Manual



```
1
2 // ScrollBarExample.java
3 // A quick demonstration of JScrollBar (both vertical and horizontal).
4 //
5 import java.awt.*;
6 import java.awt.event.*;
7 import javax.swing.*;
8
9 public class ScrollBarExample extends JPanel {
10
11     JLabel label;
12
13     public ScrollBarExample() {
14         super(true);
15         label=new JLabel();
16         setLayout(new BorderLayout());
17
18         JScrollBar hbar=new JScrollBar(JScrollBar.HORIZONTAL, 30, 20, 0, 300);
19         JScrollBar vbar=new JScrollBar(JScrollBar.VERTICAL, 30, 40, 0, 300);
20
21         hbar.setUnitIncrement(2);
22         hbar.setBlockIncrement(1);
23
24         hbar.addAdjustmentListener(new MyAdjustmentListener());
25         vbar.addAdjustmentListener(new MyAdjustmentListener());
26
27     }
```

Continued...

Lab Manual


Run as -> Java Application



Assignments To Solve

1. Design below form displaying basic swing controls.

Lab Manual



WELCOME

TextField

TextArea

.....

☒ Core Java ☐ Adv Java ☒ Male ☐ Female

AA
BB
CC
DD

AAAA

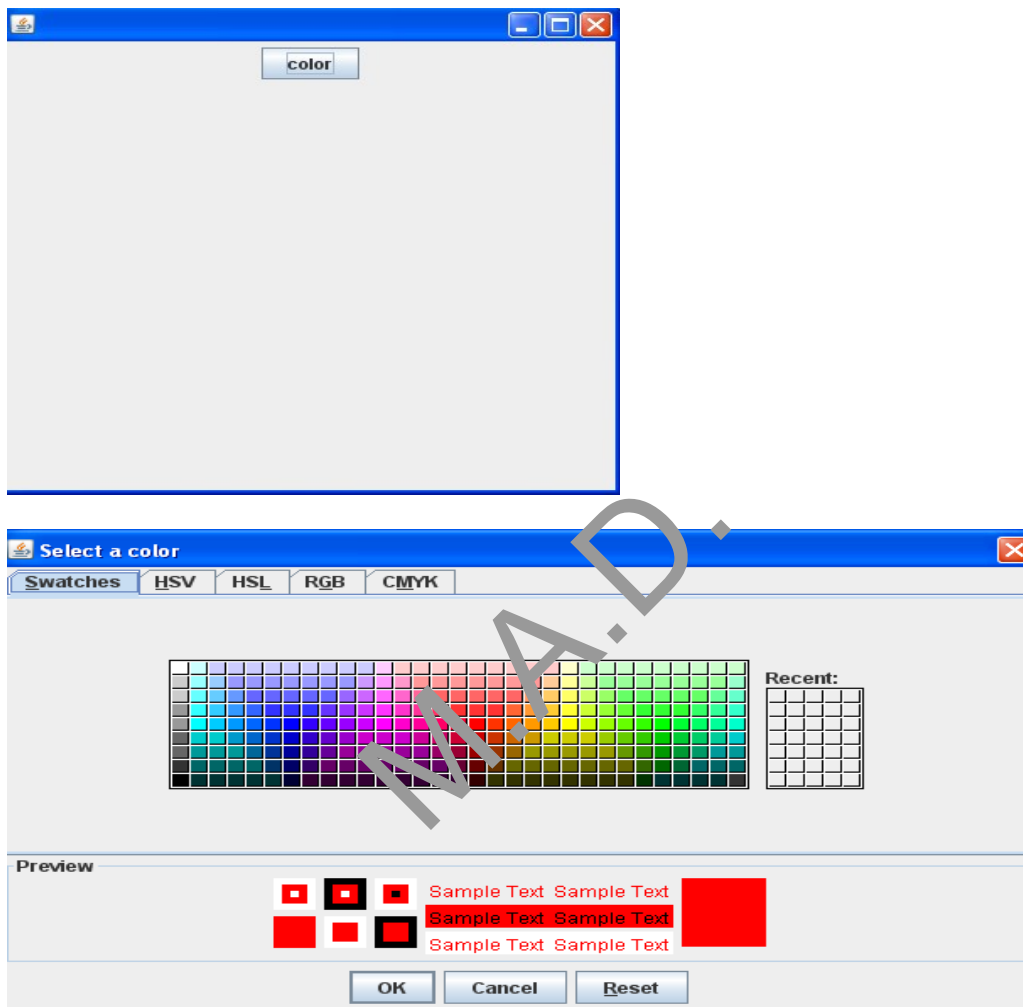
▼

Submit

M.A.D.

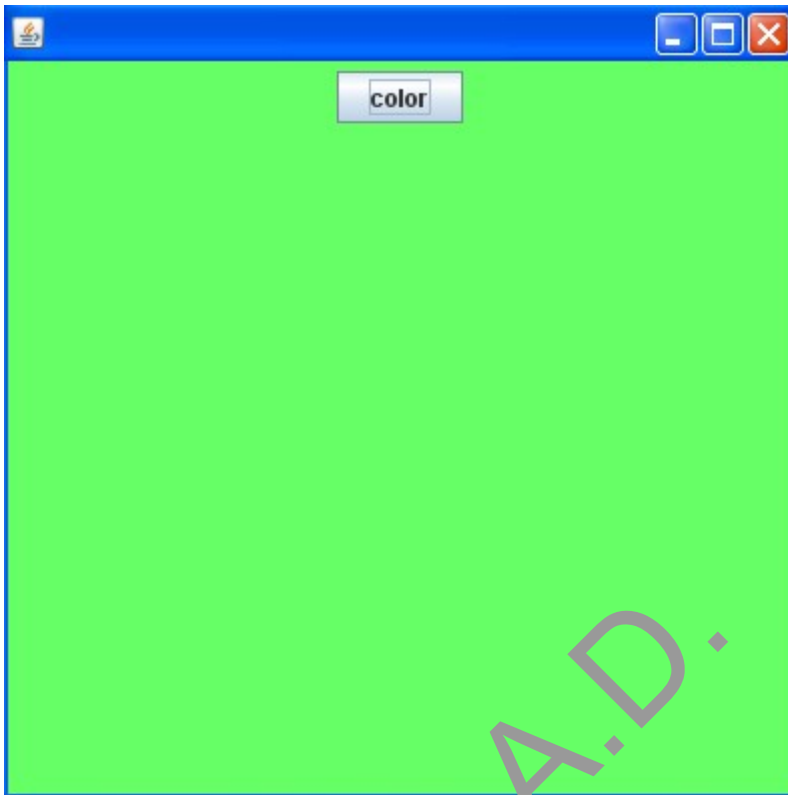
Lab Manual

2. WAP to handle an event for a button to set the background color of the frame by selecting color from JColorChooser class.

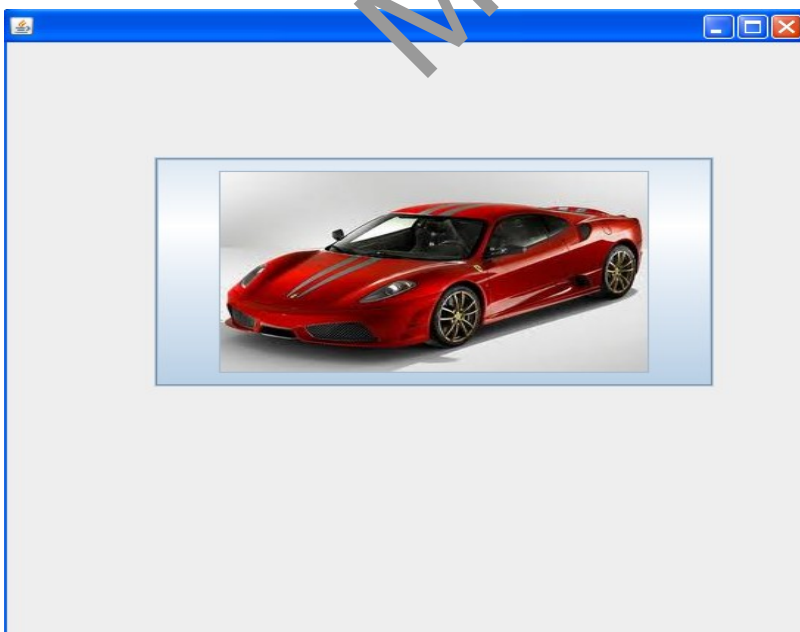


Select and press ok

Lab Manual

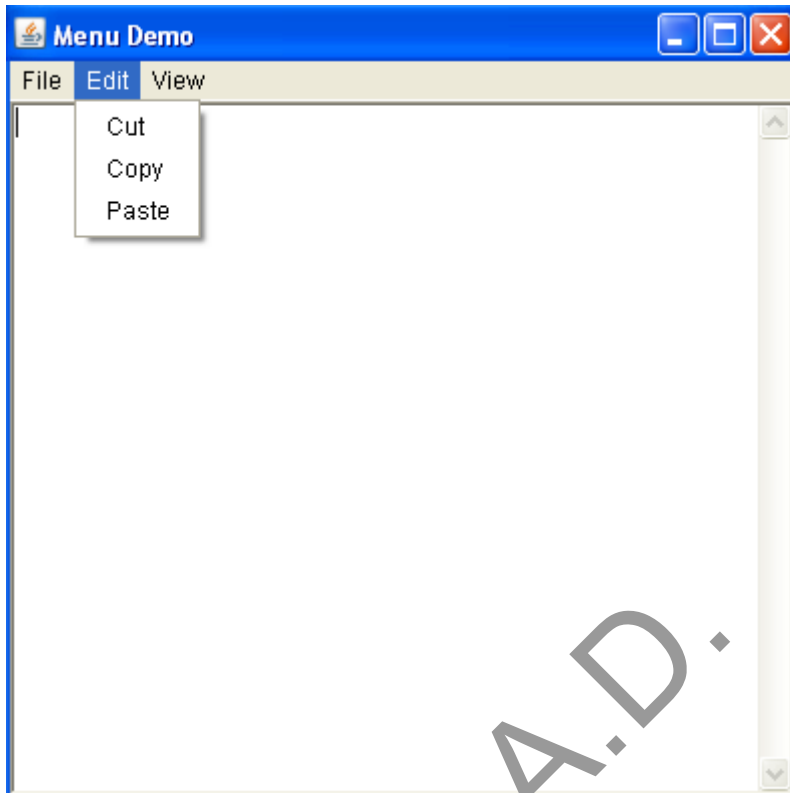


3. Write an event driven program to move the car up, down, left and right (KeyEvent)



4. Write a java program to design notepad.(Use Jmenubar,Jmenu,JmenuItem,JtextArea)

Lab Manual



Lab Manual

Lab 10

IO programming or File handling

1. Write java program to print file specification such as isFile,isDirectory,last modified date,file size,file patch etc
2. Write a java program to read & write the content to and from “myFile.txt” using FileInputStream and FileOutputStream API
3. Java program to read content from one file and write it into another file.
(Webdirect)
4. Write a java program that takes a file name and a search string from the user, if the search string occurs in the file, then it counts the no. Of occurrences of the string *(Accenture)*
Ex: output :Enter a file name : test.txt
Enter a word : you
5. Write a java program to read console data using BufferedReader API
6. Write a java program to read a file, and print 5 most frequent words having length greater than 10. *(Culture Machine)*
7. Write a java program to serialize the Student objects and deserialize it(Serializaion)

Lab 11

Collections & Generics

Collection APIs

List :ArrayList,LinkedList, Vector

Set :HashSet,LinkedHashSet,TreeSet

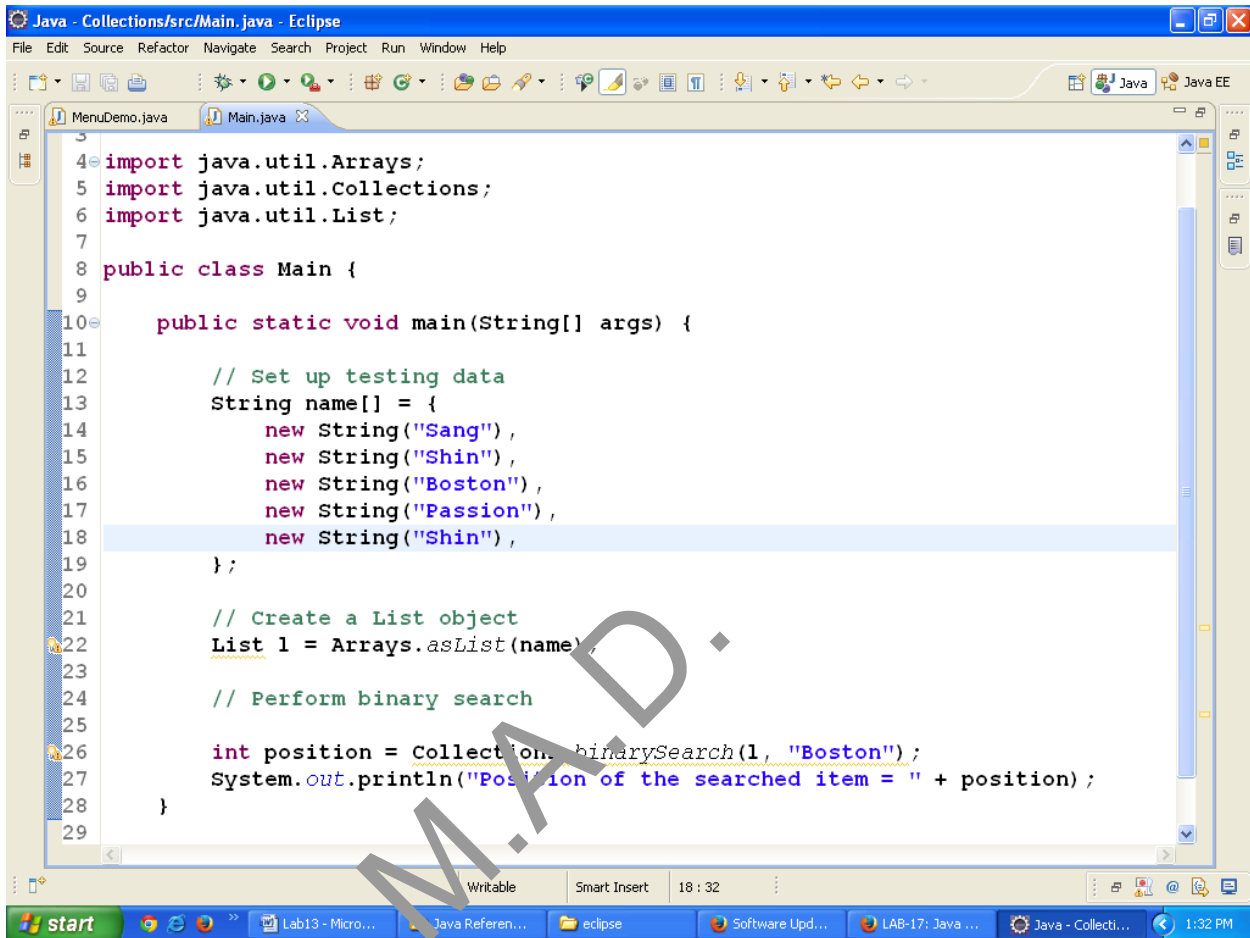
Map : HashMap,LinkedHashMap, TreeMap, HashTable

Comparable

Comparator

Demo 1:List

Lab Manual



```
Java - Collections/src/Main.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

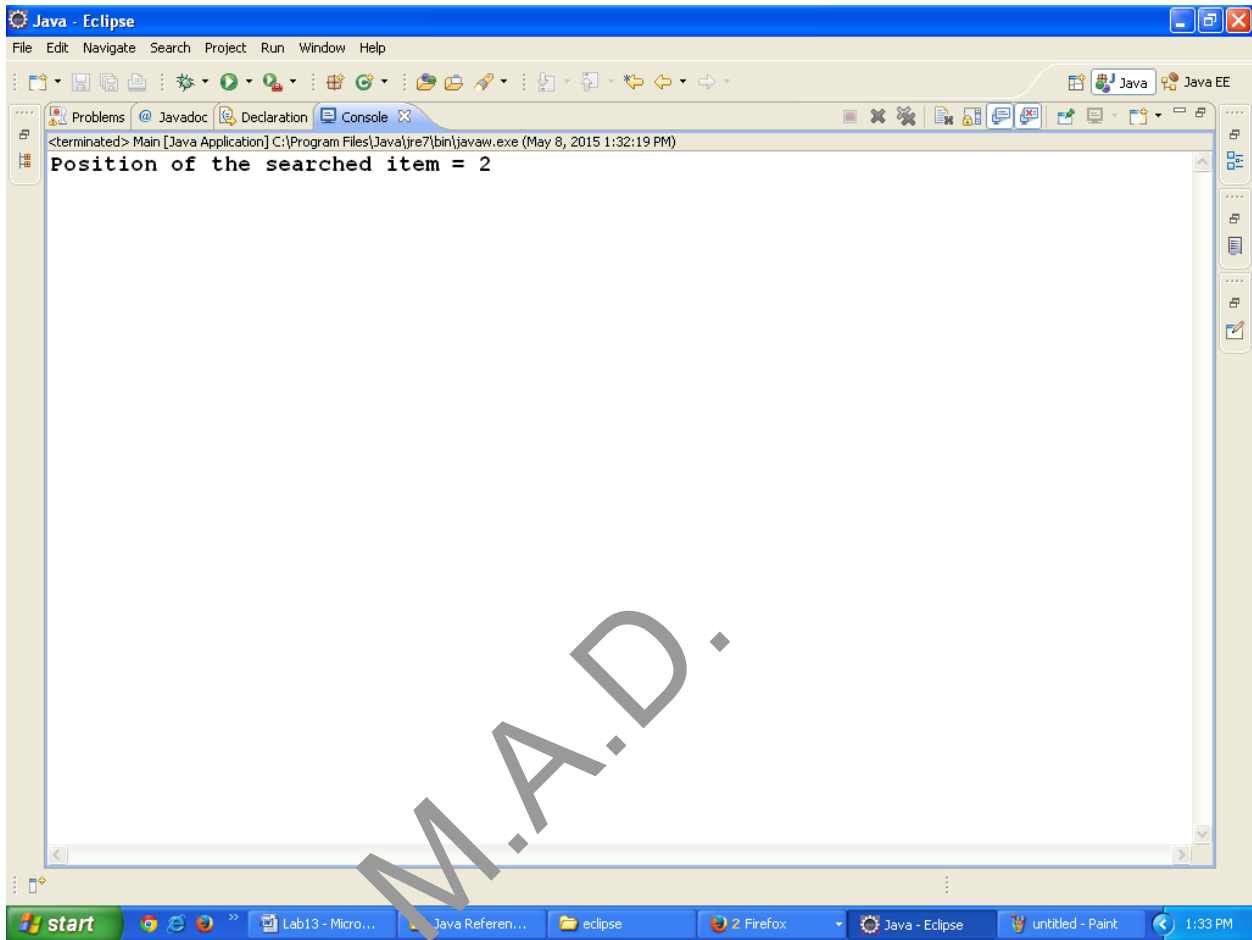
MenuDemo.java Main.java
3
4 import java.util.Arrays;
5 import java.util.Collections;
6 import java.util.List;
7
8 public class Main {
9
10     public static void main(String[] args) {
11
12         // Set up testing data
13         String name[] = {
14             new String("Sang"),
15             new String("Shin"),
16             new String("Boston"),
17             new String("Passion"),
18             new String("Shin"),
19         };
20
21         // Create a List object
22         List l = Arrays.asList(name);
23
24         // Perform binary search
25
26         int position = Collections.binarySearch(l, "Boston");
27         System.out.println("Position of the searched item = " + position);
28     }
29 }
```

Writable Smart Insert 18 : 32

start Lab13 - Micro... Java Referen... eclipse Software Upd... LAB-17: Java... Java - Collecti... 1:32 PM

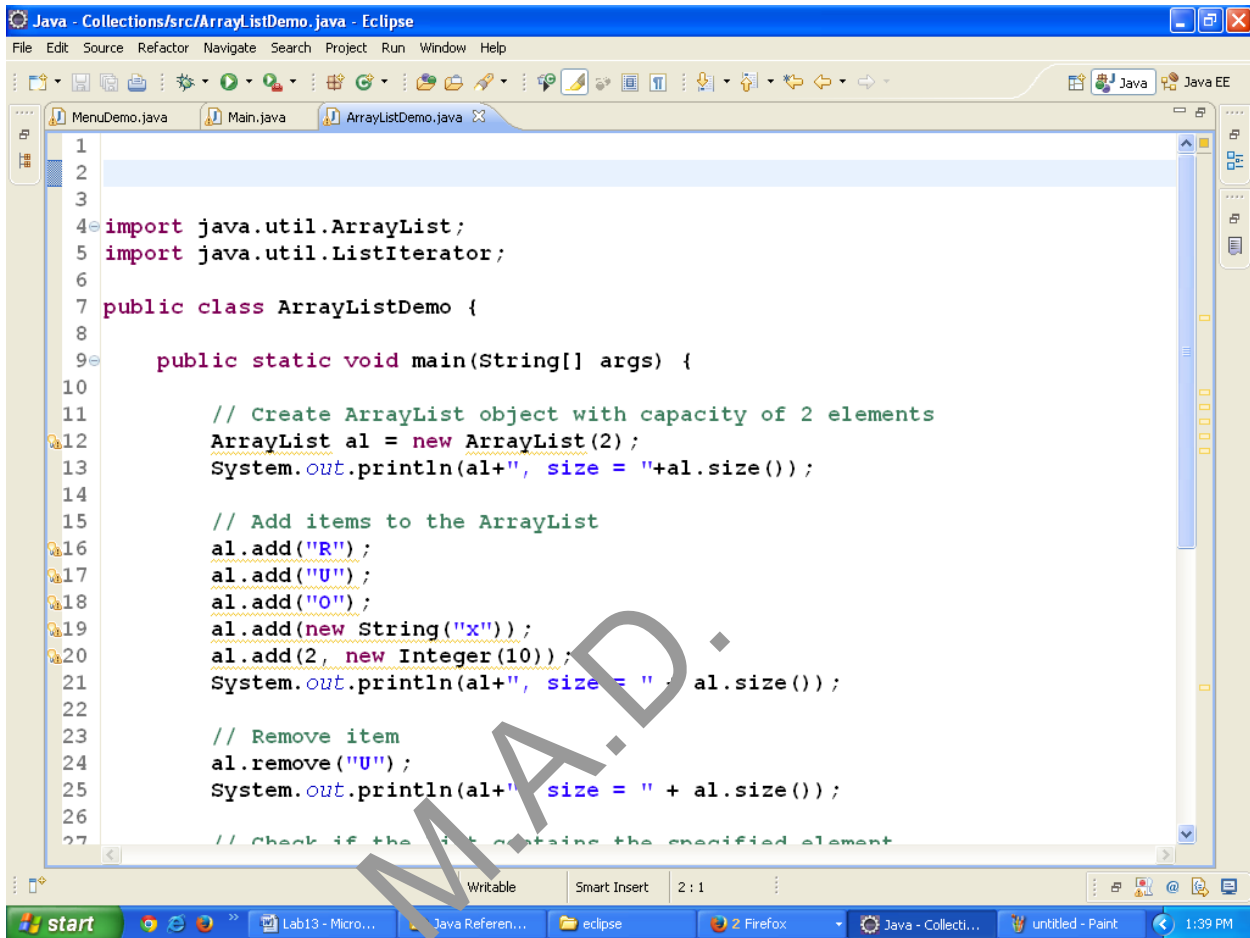
Run as -> Java Application

Lab Manual



Demo 2: ArrayListDemo

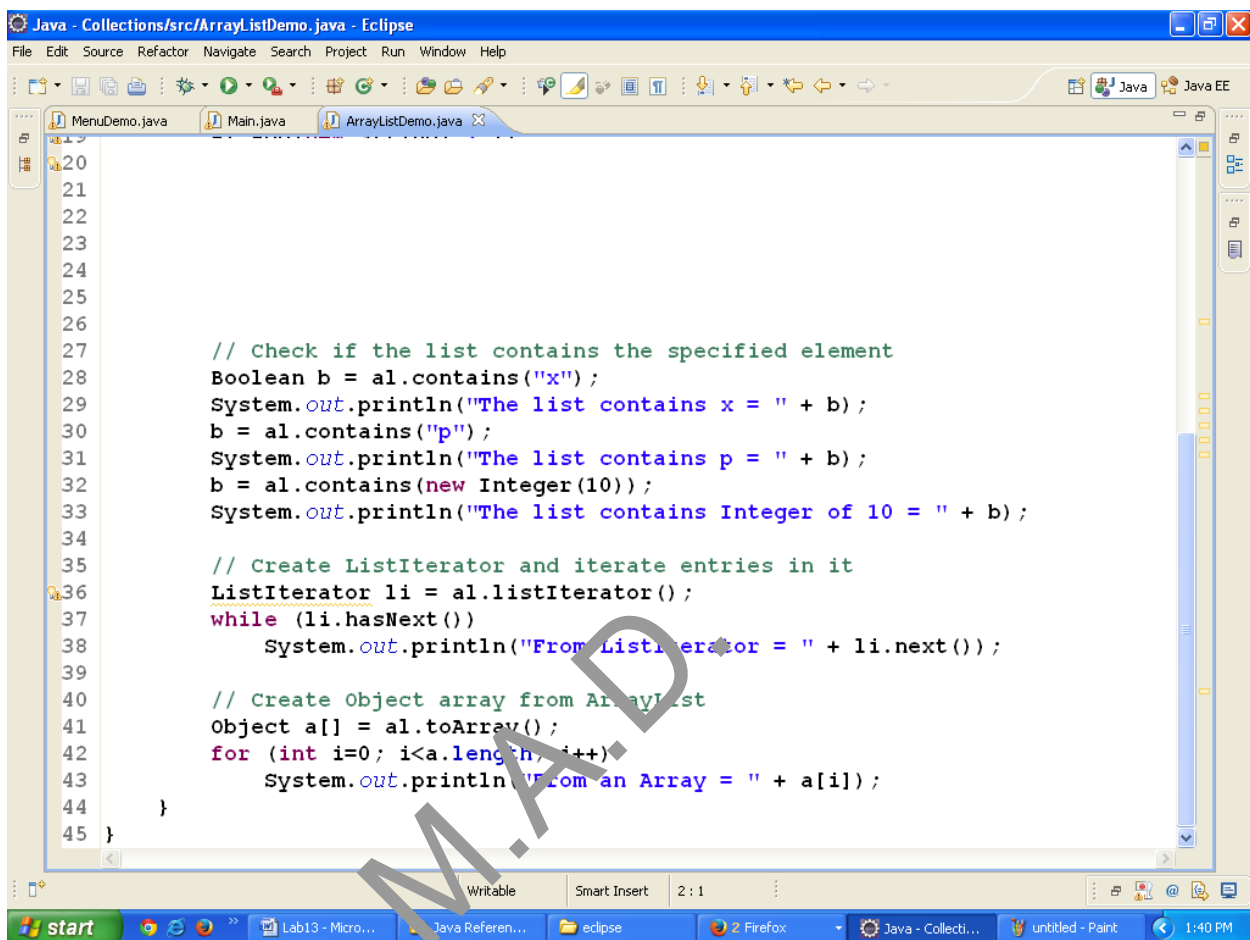
Lab Manual



```
1
2
3
4 import java.util.ArrayList;
5 import java.util.ListIterator;
6
7 public class ArrayListDemo {
8
9     public static void main(String[] args) {
10
11         // Create ArrayList object with capacity of 2 elements
12         ArrayList al = new ArrayList(2);
13         System.out.println(al+" , size = "+al.size());
14
15         // Add items to the ArrayList
16         al.add("R");
17         al.add("U");
18         al.add("O");
19         al.add(new String("x"));
20         al.add(2, new Integer(10));
21         System.out.println(al+" , size = "+al.size());
22
23         // Remove item
24         al.remove("U");
25         System.out.println(al+" , size = "+al.size());
26
27         // Check if the list contains the specified element
```

Continued...

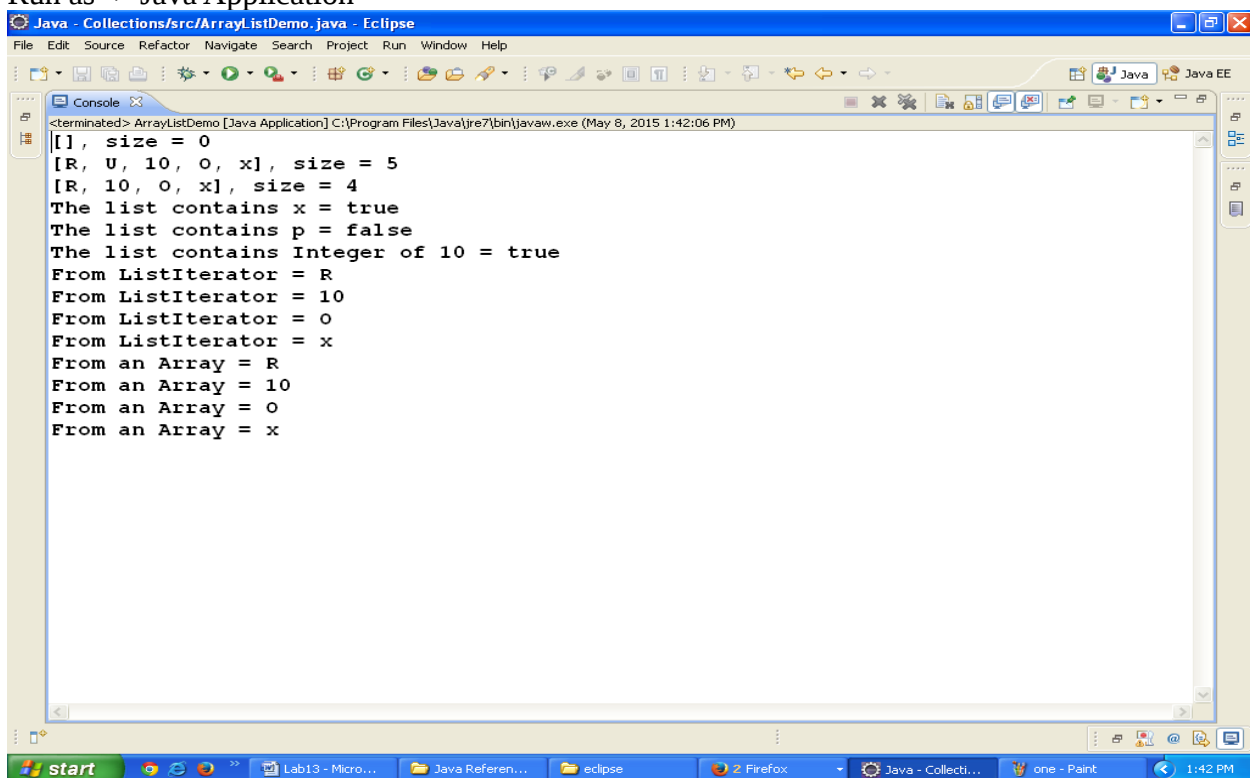
Lab Manual



The screenshot shows the Eclipse IDE with the file `ArrayListDemo.java` open. The code is as follows:

```
20
21
22
23
24
25
26
27 // Check if the list contains the specified element
28 Boolean b = al.contains("x");
29 System.out.println("The list contains x = " + b);
30 b = al.contains("p");
31 System.out.println("The list contains p = " + b);
32 b = al.contains(new Integer(10));
33 System.out.println("The list contains Integer of 10 = " + b);
34
35 // Create ListIterator and iterate entries in it
36 ListIterator li = al.listIterator();
37 while (li.hasNext())
38     System.out.println("From ListIterator = " + li.next());
39
40 // Create Object array from ArrayList
41 Object a[] = al.toArray();
42 for (int i=0; i<a.length; i++)
43     System.out.println("From an Array = " + a[i]);
44 }
45 }
```

Run as -> Java Application



The screenshot shows the Eclipse IDE with the console window open, displaying the output of the Java application. The output is as follows:

```
<terminated> ArrayListDemo [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 8, 2015 1:42:06 PM)
[[], size = 0
[R, U, 10, O, x], size = 5
[R, 10, O, x], size = 4
The list contains x = true
The list contains p = false
The list contains Integer of 10 = true
From ListIterator = R
From ListIterator = 10
From ListIterator = O
From ListIterator = x
From an Array = R
From an Array = 10
From an Array = O
From an Array = x
```

Lab Manual

M.A.D.

Assignments To Solve :

1. Write a program to test add(),get(),size(),isEmpty(),iterator(),remove() functions with basic collection APIs(List,Set)

2. Write a Menu Driven program to do following

1. To Add elements into ArrayList
2. To retrieve an element (at a specified index) from a given array list
3. To insert an element into the array list at the first position
4. To update specific array element by given element.
5. To remove element from given position
6. To Search element in an arraylist
7. To reverse element of an arraylist
8. To sort element of an arraylist

3. WAP to generate the following GUI



Handle event for Add, List All and Search Buttons
Event for Add:

Retrieve the elements from the textfields, create an Object of Employee and add the Objects to Vector or ArrayList.

Lab Manual

Event for List All

Display All Objects in TextArea

Event for Search

Iterate through the collection to find employee object for a particular id entered and display the details of that Object in TextArea..Or else print message Object not found for given id..

The screenshot shows a Java Swing window with a light gray background and a blue title bar. It contains a form with three labeled text fields: 'ID' (containing '1002'), 'Name' (containing 'Sumit'), and 'Basic' (containing '85000'). Below these fields are two buttons: 'Add' and 'List All'. A text area in the center displays the text: 'Collection : [ID =1001 Name=Harsh Basic=9000, ID =1002 Name=Sumit Basic=85000]'. At the bottom, there is a 'Search' button and a text field containing '1001'.

4. SMS language is the term for the abbreviation and its commonly used with mobile phone text messaging (use collection framework)(*Vyom Labs*) some of the abbreviation used are

s for yes	2day for today
u for you	y for why

Write a program that would convert a given text into proper SMS language
 Example :
 Input :Where were you yesterday
 Output :where were u sterday.

Lab Manual

5. A program that prints even and odd characters from a string. (*Consult Add*)

5. Create a Product class with Product Id & Product Name. Write a program to accept information of 10 products and store that in HashTable. Search a particular product in the Hash Table. Remove a particular product id and product name from the Hash Table.

The product list is as follows:

Product Id	Product Name
P001	Maruti800
P002	MarutiZen
P003	MarutiEsteem

6. Write a Java program to Store the <telephone_no,Name_of_Customer> and sort the data by key of descending order

7. Write a Java program which take every entry as name and age of student, and save it as below specified way (*BMC*)

Example :

Input :

Enter name and age of 1th :

Pravin 25

Enter name and age of 1th :

Prmod 20

Enter name and age of 1th :

Soham 25

Enter name and age of 1th :

Pravin 5

O/P:

Pravin 25,5

Prmod 20

Soham 25

Lab Manual

Lab 12

MultiThreading

Thread class and their methods

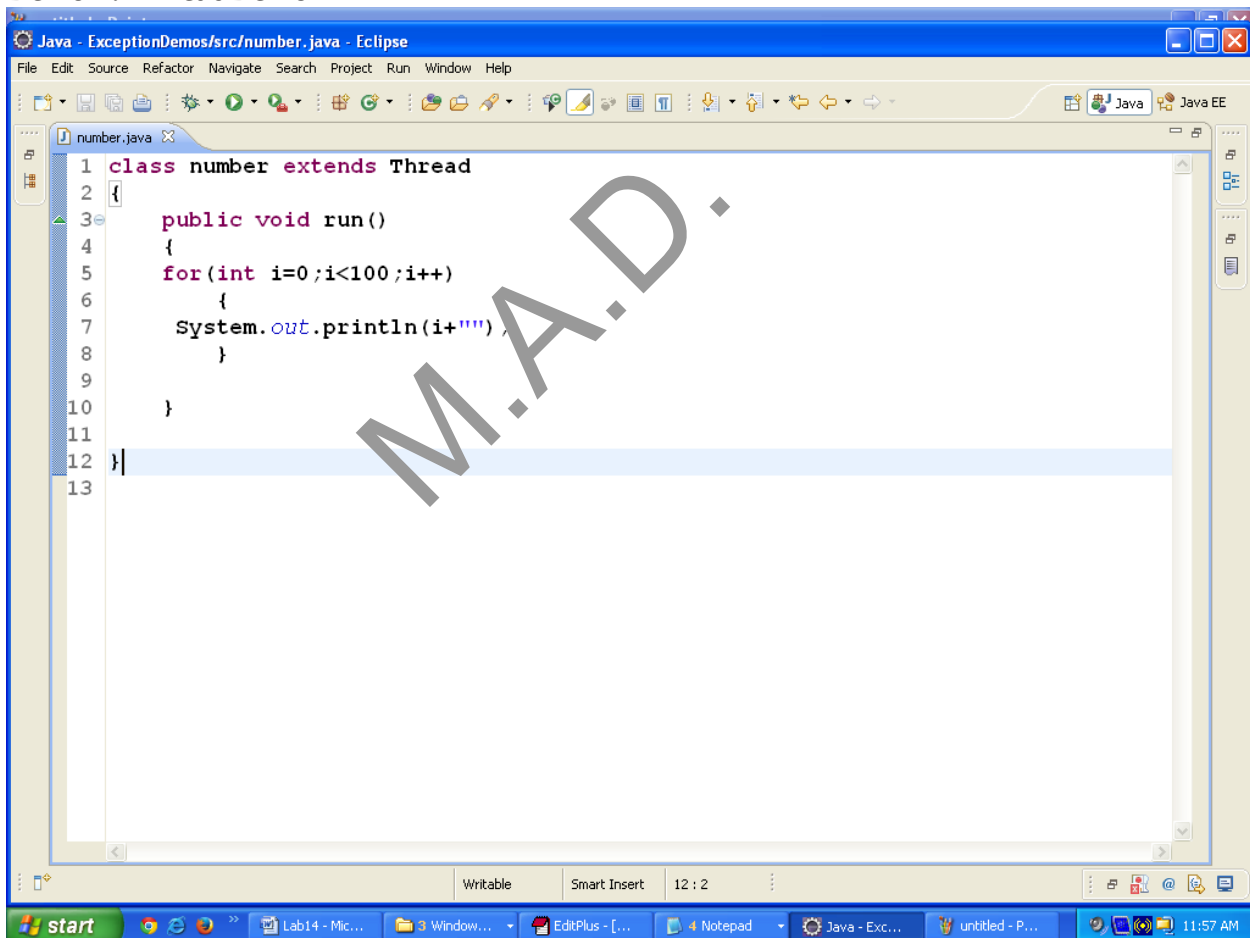
Runnable Interface

Synchronization

Interthread communication

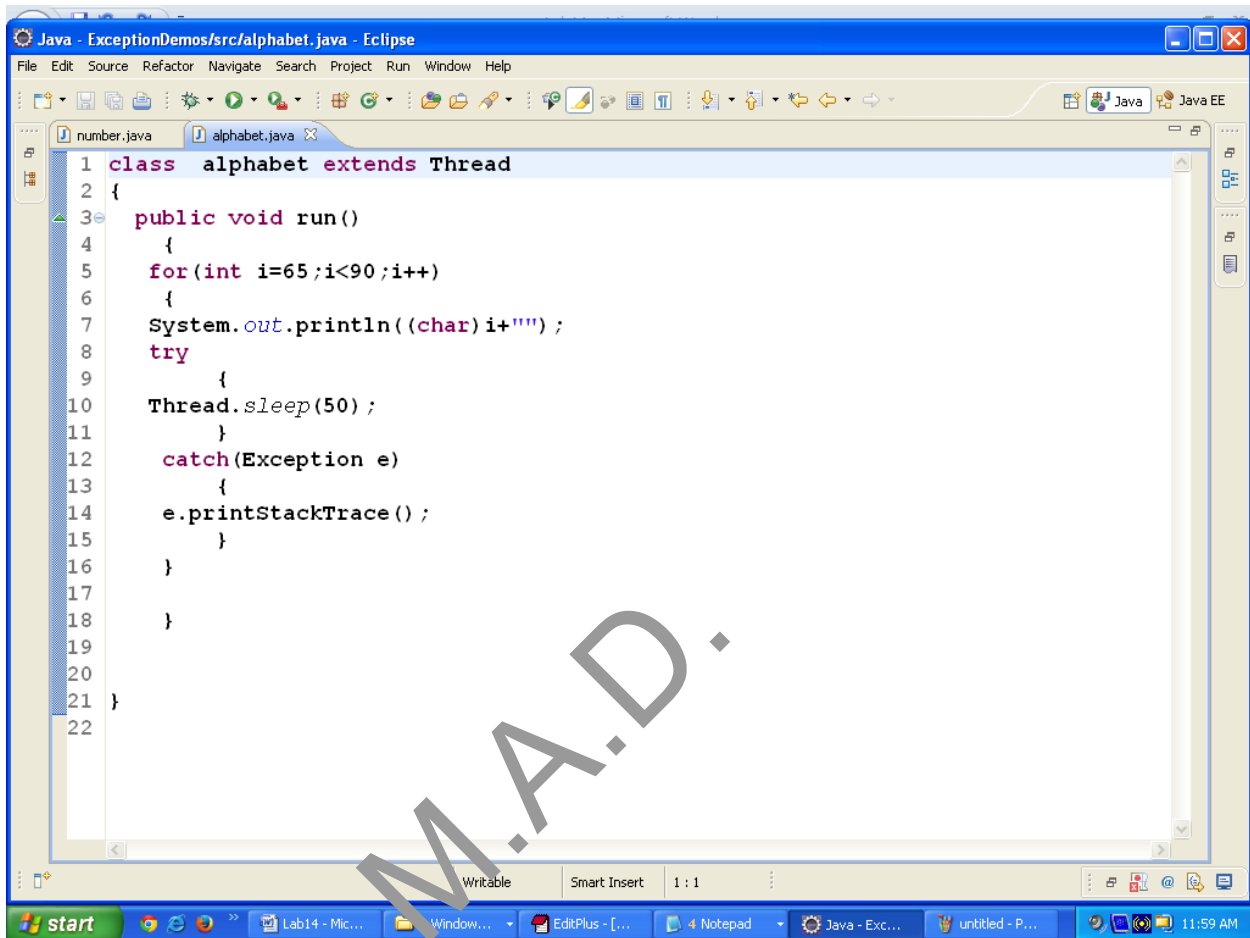
Lock api

Demo 1: ThreadDemo

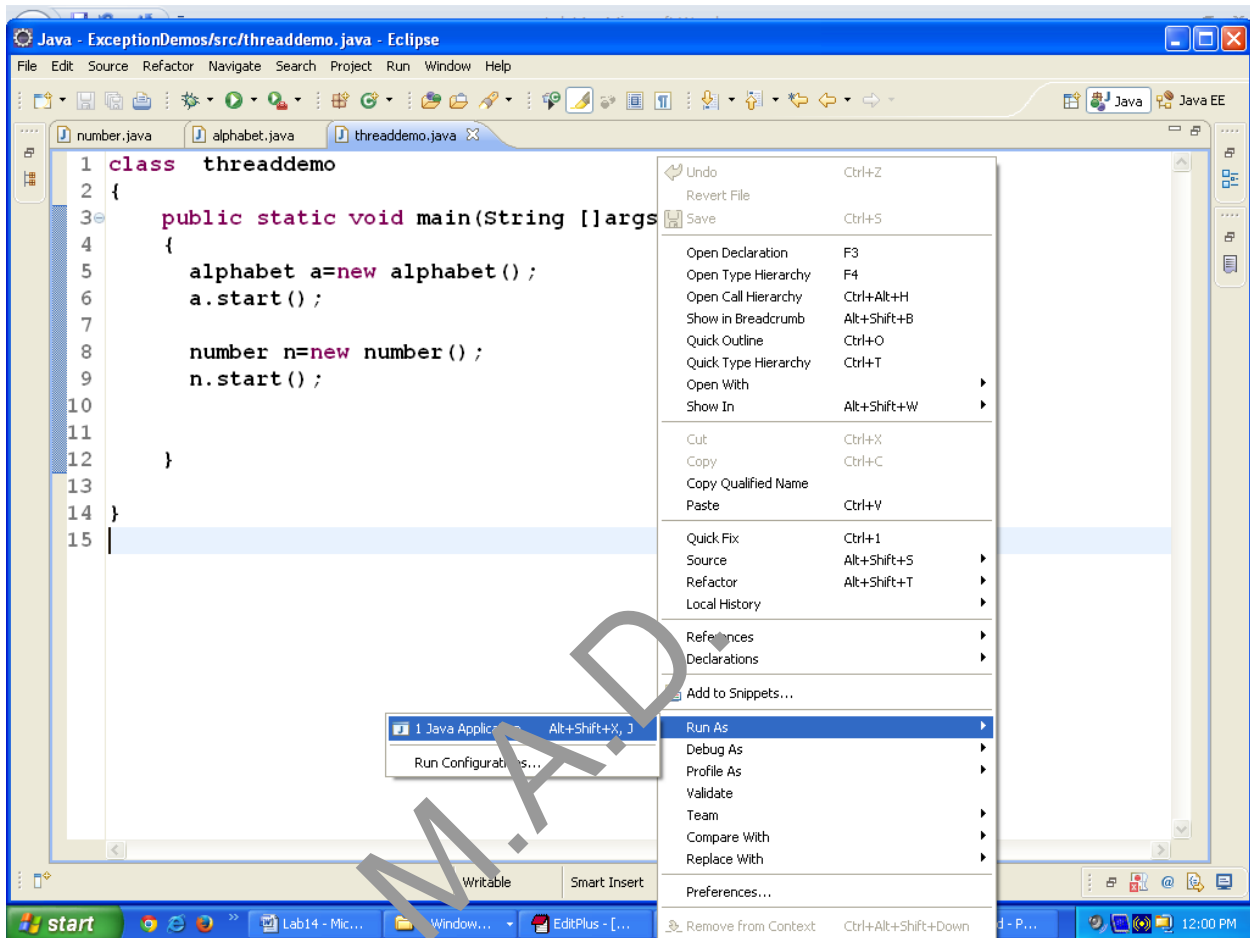


```
1 class number extends Thread
2 {
3     public void run()
4     {
5         for(int i=0;i<100;i++)
6         {
7             System.out.println(i+"");
8         }
9     }
10 }
11
12 }
13
```

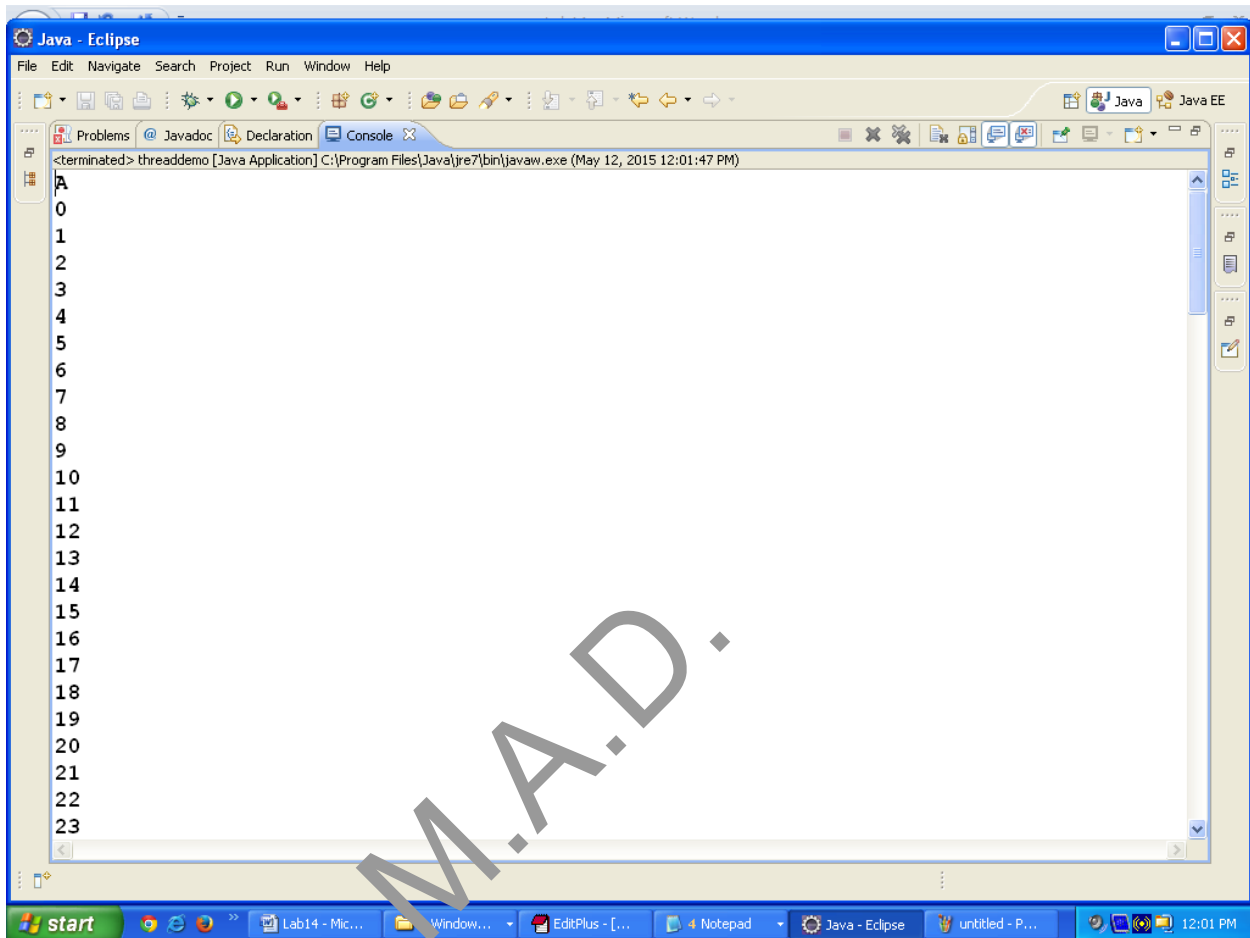
Lab Manual



Lab Manual



Lab Manual



Assignments To Solve

1. Write a Java program to create Thread and practice below methods.

getName(): It is used for Obtaining a thread's name

getPriority(): Obtain a thread's priority

isAlive(): Determine if a thread is still running

join(): Wait for a thread to terminate

run(): Entry point for the thread

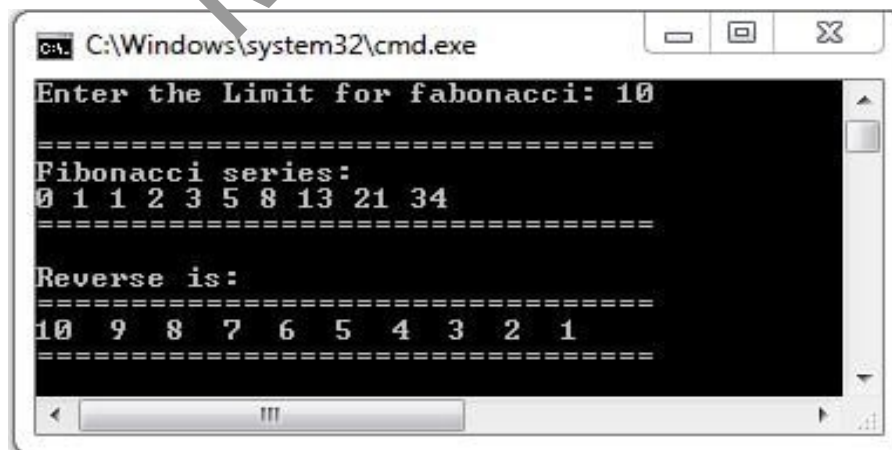
sleep(): suspend a thread for a period of time

start(): start a thread by calling its run() method

2. Write a JAVA program which will generate the threads:

- To display 10 terms of Fibonacci series. (class Fibonacci extends Thread)
- To display 1 to 10 in reverse order. (class Reverse extends Thread)

Output :



```
C:\Windows\system32\cmd.exe
Enter the Limit for fabonacci: 10
=====
Fibonacci series:
0 1 1 2 3 5 8 13 21 34
=====
Reverse is:
10 9 8 7 6 5 4 3 2 1
=====
```

3. Create a Banking Application simulating concurrent deposit and withdrawal. Use synchronized method for transaction.

Account Class

Attributes: acno & balance

Methods: getter setter methods and toString()

Lab Manual

Method signature

```
public synchronized int withdraw(Account acc, int amount)
{....}
```

```
public synchronized int deposit(Account acc, int amount)
{.....}
```

4. Create a Banking Application simulating concurrent deposit and withdrawal.
Use synchronized block for transaction

Method signature

```
public int withdraw (Account acc, int amount)
{
synchronized (this)
{.....}
}
```

```
public int deposit(Account account, int amount)
{
synchronized {....}
}
```

5. WAP to move three balls (Use wait(), notify() and notifyAll() of Object class)
6. Write a java program to demonstrate explicit wait condition check (**Cognizent**)

Lab 13

1.JDBC Api

Connection**Statement****PreparedStatement****ResultSet**

Assignments To Solve:

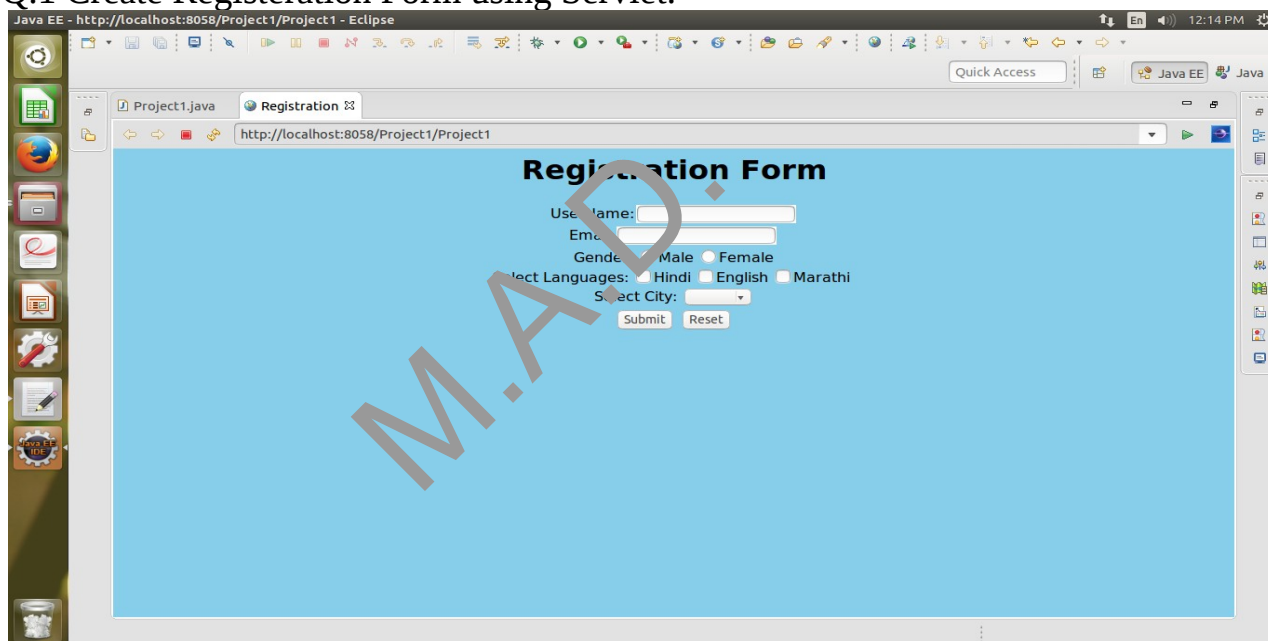
- 1.WAP perform CRUD operation on Student(rollno,name,marks,address) table using JDBC APIs .(use Statement)

Lab Manual

2. WAP perform CRUD operation on Book(id,name,author,price,category) table using JDBC APIs .(use PreparedStatement)
3. WAP to perform CRUD operation On Employee Model (use collection as database) (**HSBC**)

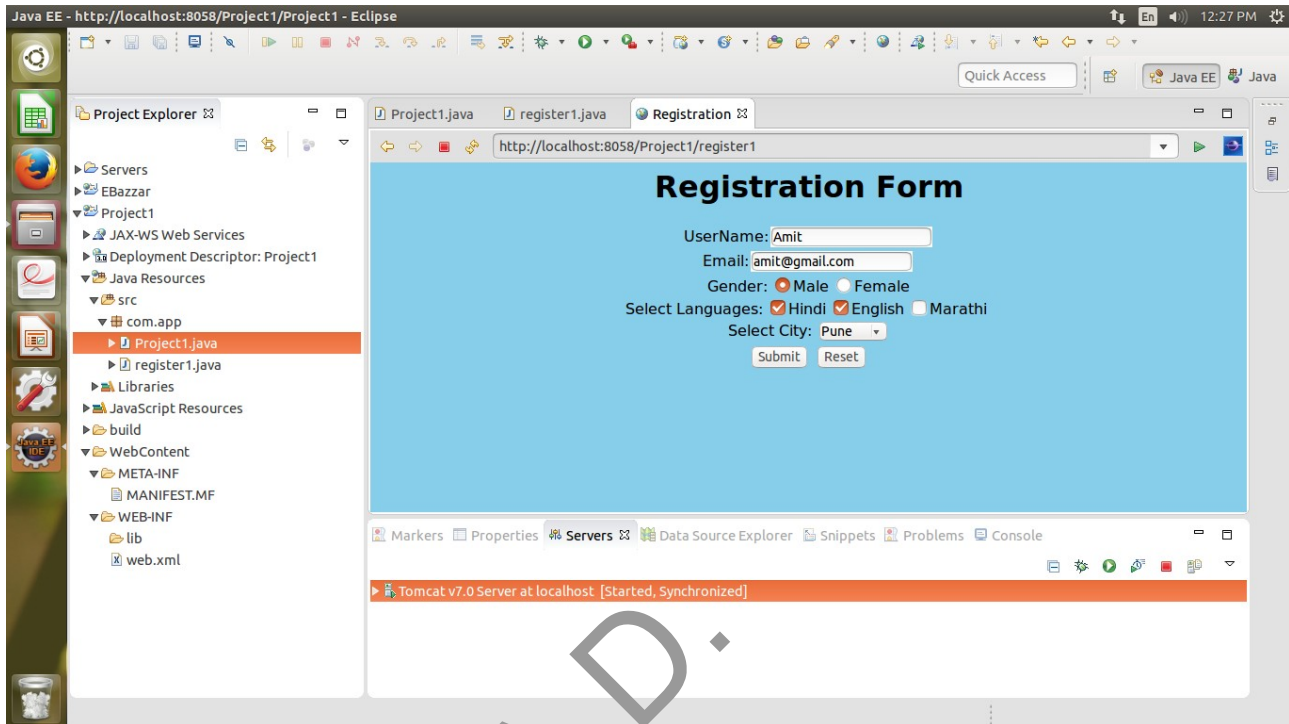
Servlet Assignments

Q.1 Create Registration Form using Servlet.



and Display in Next Servlet.

Lab Manual



Q.2 Write a servlet that will take a string and show the reverse in the form. (create three servlets, FirstServlet take text input and submit it to SecondServlet,

The SecondServlet print that input in reverse, it has original button when click, it navigate to ThirdServlet and then

ThirdServlet print original text
)



Lab Manual



Feed in some text:

Press "submit" and get the form back with the text reversed:



Lab Manual



Clicking "submit" again should give you the original text:
Using Servlet basic API.

Q.3 Store email in web.xml as init-param name value pair. Write a Servlet to extract the initialization parameters inside the init method of your Servlet and display it within your service method.

Q.4 Store database specification into web.xml as context-param name value pair. Write a Servlet to extract the context parameters ,and display it within your service method.

Q5. Develop LoginApp as directed below flow

Login.html[accept username , password and registration link]

ValidateServlet (check username and password is valid or not with database,depends on that it will redirect to either HomeServlet or Login.html)

When you click on register link it will redirect you to Register.html and take all registration details, after submit it will save all detail using RegisterServlet

Q.6 Create web application to pay online telephone bill using
(Servlet,ServletConfig,ServletContext & RequestDispatcher with Database.)

Hint:

Database BSNL

Table PayBill having Cust_No,Cust_Name & Current_Bill

Servlet

Login.html

User enter contact number and login

Lab Manual

DisplayServlet.java

Display CustomerName & CurrentBill which fetch from DB using ServletConfig & ServletContext.

Click on “Pay” button.

Pay.java

Display Bill payment successful with DB update.

Else

Back to Login if contact number is incorrect.

Q.7 Display Customer BankDetails like Name, CustomerId

Account_no using ServletFilter chaining concept.

CustomFilter—filter customerID

AuthenticateFilter – filter Account_no.

Q.8 Develop BookCart Application using Servlet concept in incremental approach.

Scenario:

Cyber Publishing Company is a popular publishing company in India. All the transactions of books happen manually till now. They want to sell books online due to increase No. of customers.

They need to develop an web based application to purchase books online.

Login Id <Login Id >		Category <Category >		
Book Id	Book Name	Author Name	Price	Purchase
---	----	----	----	<input type="checkbox"/>
----	---	----	----	<input type="checkbox"/>
	Submit [Hyperlink]	Logout [Hyperlink] [Hyperlink]	Show Cart	

5. Login ID must be taken from login page.
6. Book Category must be either Computer or Engg Books.
7. If Computer Books selected then display all the Books with all the details.
8. If Engg. Books selected then display all the Books with all the details.
9. Click on Logout Hyperlink will display the page with Thanks message

Lab Manual

and a hyperlink which will move the pointer to the login page again.

6. The customer needs to select the checkbox corresponding to **Purchase** if he is interested in that Book.
 - Once the customer has selected all the Books for Purchase he needs to click the **Submit** Button.
 - On click of the Hyperlink **Show Cart** the **ShowCartPage** will be called.

The screen of **ShowCartPage** is as follows:

Login Id		<Login Id >	
Book Id	Book Name	Price	Amount
-----	-----	----	100
-----	-----	----	200
		Total Amount	300

JSP Assignments

Q.1 Write JSP Script which accepts user name and nick name from user. At first visit, display message “Hello user name” and for next successive requests, display “Hello nick name”.

Use username if visit count is odd and nick name if visit count is even. (use declaration

scripting elements)

Q.2 Write jsp code to use page directives tag's **isErrorPage**, **errorPage**, **import** attribute.

Q.3 Write a jsp code to use all jsp implicit object to read and set content with different scope.

Q.4 Using Scripting Element & Directive tag.

Problem Statement 1:

ZBank is new generation bank they are planning to launch their website. They have approached you to create the banking application using jsp. As initial task to test your expertise in jsp programming they have asked you to create a simple welcome screen with a greeting message.

Requirement Description:

The welcome page should be named welcome.jsp. It should have the current date displayed “**dd-MM-yyyy**” format. The page should display a greeting based on the time of the day.

if the time is less than 12.00

Print “Good Morning”

Else Print “Good Evening”

The bank wishes to use 24hr time format.

Also as a matter of initial promo the bank wishes to keep track of the number of users visiting the site and this value should be displayed to the user.

Lab Manual

Develop a jsp page satisfying the above requirements.

The page should look like the screen shot design provided and the following test cases also needs to be executed.

Hints:

4. Use the following code for date calculations and use scriptlets

```
java.util.Calendar calendar = new java.util.GregorianCalendar();
```

```
java.util.Date date = calendar.getTime();
```

```
DateFormat format=new SimpleDateFormat("dd-MM-yyyy");
```

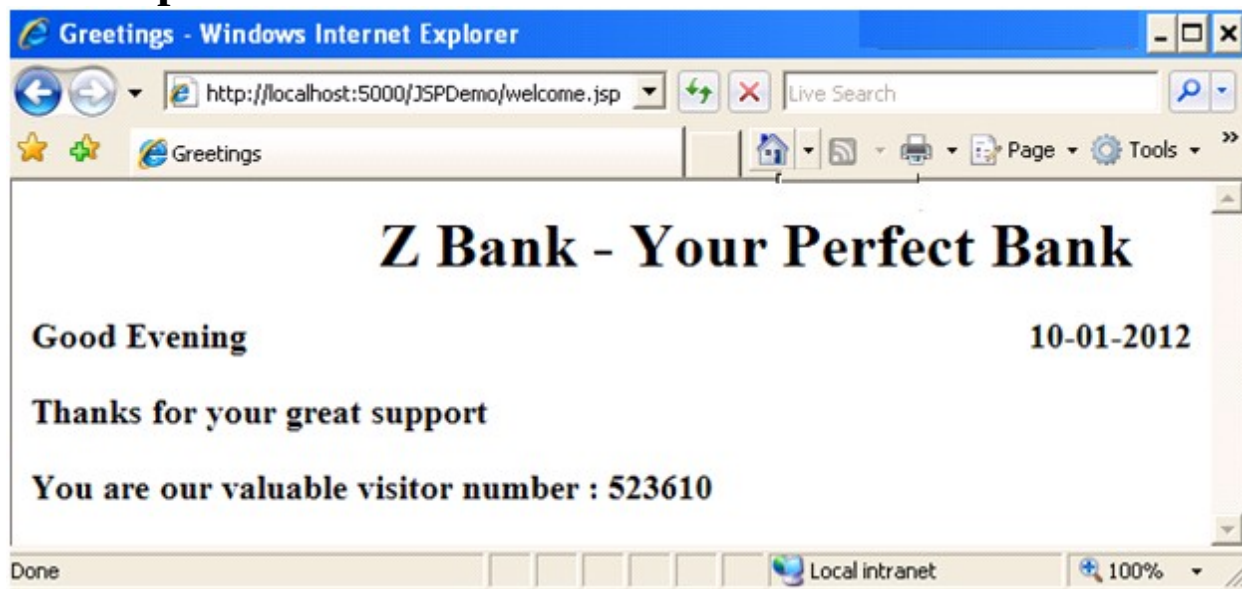
```
String dateval=format.format(date);
```

```
int hour = calendar.get(Calendar.HOUR_OF_DAY);
```

5. Use declaration to declare page count variable.

Test Cases: Execute the below test cases after development.

10. **Test case #1:** Access the welcome.jsp it should display the current system date and display the appropriate message.
11. **Test case #2:** As you refresh the JSP page the counter needs to be incremented and should increase for each refresh.

Screen Expected

Q.5 Write JSP code which adds the student education details in a database through Java bean.

Also display the student details who have secured first class in their graduation.

Lab Manual

Q6. Write JSP code to perform CRUD operation on Product table.
Define links into index.html (add new Product, display products etc)

Q.7 Create custom tag which display User Name in its initial.
For eg Hrithik Rakesh Roshan ---> H.R.Roshan using either tag with body or attribute.

Q.8 Create custom tag which display UserName in uppercase with specified font color
(use tag with body and attribute)

M.A.D.