

## CS8383 –OBJECT ORIENTED LABORATORY SYLLABUS

### COURSE OBJECTIVES

- To build software development skills using java programming for real-world applications.
- To understand and apply the concepts of classes, packages, interfaces, arraylist, exception handling and file processing.
- To develop applications using generic programming and event handling.

### LIST OF EXPERIMENTS:

1. Develop a Java application to generate Electricity bill. Create a class with the following members:  
Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 per unit
- 101-200 units - Rs. 2.50 per unit
- 201 -500 units - Rs. 4 per unit
- > 501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 per unit
- 101-200 units - Rs. 4.50 per unit
- 201 -500 units - Rs. 6 per unit
- > 501 units - Rs. 7 per unit

2. Develop a javaL application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa) , time converter (hours to minutes, seconds and vice versa) using packages.

3. Develop a java application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.
4. Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations.
5. Write a program to perform string operations using ArrayList. Write functions for the following
  - a. Append - add at end
  - b. Insert – add at particular index
  - c. Search
  - d. List all string starts with given letter
6. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
7. Write a Java program to implement user defined exception handling.
8. Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.
9. Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
10. Write a java program to find the maximum value from the given type of elements using a generic function.
11. Design a calculator using event-driven programming paradigm of Java with the following options.

- a) Decimal manipulations
- b) Scientific manipulations

12. Develop a mini project for any application using Java concepts.

[www.LearnEngineering.in](http://www.LearnEngineering.in)

**CS8383 – OBJECT ORIENTED PROGRAMMING LABORATORY**  
**INDEX**

Sl. No.	Name of the Experiment	Page No.
1	Electricity Bill Generation	13
2	Currency converter , Distance converter and Time converter	15
3	Payroll Processing	18
4	ADT Stack	19
5	String Operations	21
6	Abstract Class	23
7	Exception Handling	25
8	File Information	27
9	Multithreading	29
10	Generic Function	31
11	Calculator	33
12	Mini Project – Vehicle Management System	36

**Ex. No.:1**

**ELECTRICITY BILL GENERATION**

**Aim:**

To develop a Java application to generate Electricity bill.

**Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

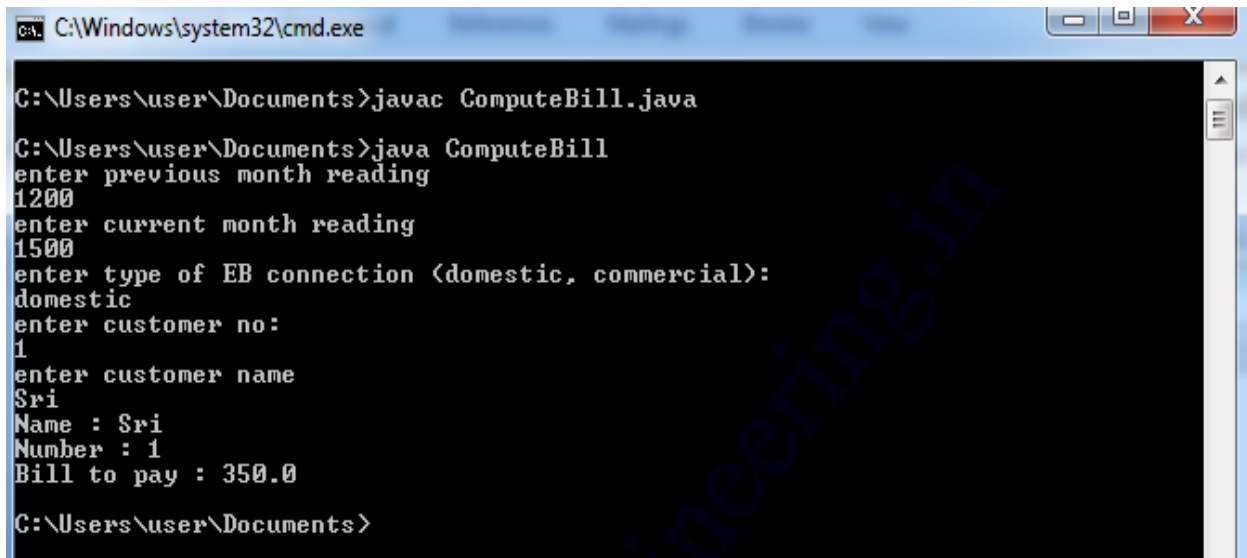
**Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

**Procedure:**

1. Start the program
2. Create consumer class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial).
- 3.If the type of the EB connection is domestic, calculate the amount to be paid as follows:
  - First 100 units - Rs.1 per unit
  - 101-200 units - Rs.2.50 per unit
  - 201 -500 units - Rs. 4 per unit
  - > 501 units - Rs.6 per unit
4. If the type of the EB connection is commercial, calculate the amount to be paid as follows:
  - First 100 units - Rs. 2 per unit
  - 101-200 units - Rs. 4.50 per unit
  - 201 -500 units - Rs. 6 per unit
  - > 501 units - Rs. 7 per unit
5. Print the amount
6. Stop the program

**Sample output:**



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

C:\Users\user\Documents>javac ComputeBill.java

C:\Users\user\Documents>java ComputeBill
enter previous month reading
1200
enter current month reading
1500
enter type of EB connection (domestic, commercial):
domestic
enter customer no:
1
enter customer name
Sri
Name : Sri
Number : 1
Bill to pay : 350.0

C:\Users\user\Documents>
```

**Result:**

Thus the Electricity Bill Java application was successfully executed.

**Outcome:**

Thus the course outcome (CO1) has been attained by generating an Electricity Bill application using Java.

**Applications :**

- (1) Payroll Processing
- (2) PF Calculation

## Ex. No.:2

### Currency Converter, Distance Converter and Time Converter

#### Aim:

To develop a Java application to implement currency converter , distance converter and time converter using packages.

#### Hardware Requirements:

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

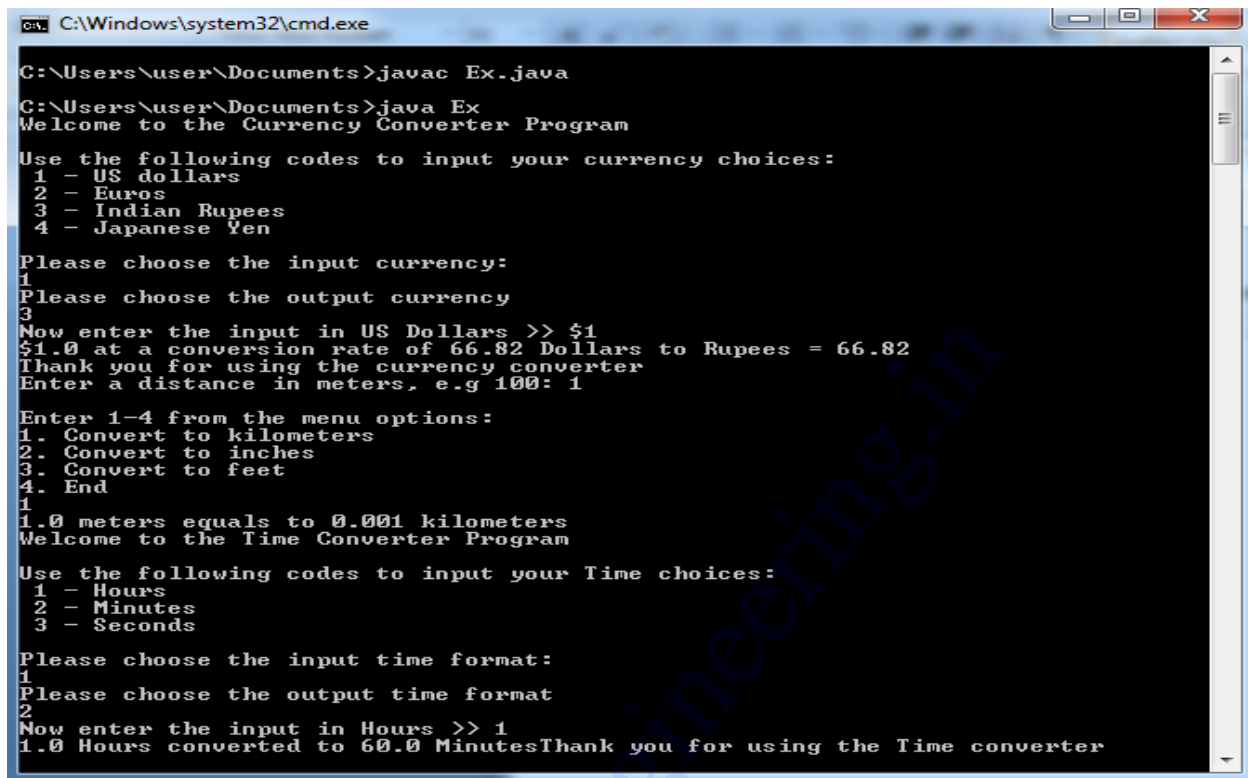
#### Software Requirements:

- Ubuntu 14.04 operating system
- JDK 1.8

#### Procedure:

1. Start the program
2. Create three packages for currency converter , distance converter and time converter.
3. Create corresponding code for conversion
4. Print the converted value.
5. Stop the program

### Sample output:



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

C:\Users\user\Documents>javac Ex.java
C:\Users\user\Documents>java Ex
Welcome to the Currency Converter Program

Use the following codes to input your currency choices:
1 - US dollars
2 - Euros
3 - Indian Rupees
4 - Japanese Yen

Please choose the input currency:
1
Please choose the output currency
3
Now enter the input in US Dollars >> $1
$1.0 at a conversion rate of 66.82 Dollars to Rupees = 66.82
Thank you for using the currency converter
Enter a distance in meters, e.g 100: 1

Enter 1-4 from the menu options:
1. Convert to kilometers
2. Convert to inches
3. Convert to feet
4. End
1
1.0 meters equals to 0.001 kilometers
Welcome to the Time Converter Program

Use the following codes to input your Time choices:
1 - Hours
2 - Minutes
3 - Seconds

Please choose the input time format:
1
Please choose the output time format
2
Now enter the input in Hours >> 1
1.0 Hours converted to 60.0 MinutesThank you for using the Time converter
```

### Result:

Thus the Java application has been created for currency conversion, distance conversion and time conversion and it was successfully executed.

### Outcome:

Thus the course outcome (CO2) has been attained by applying the concept of packages using Java.

### Applications :

- (1) Kilogram to gram conversion
- (2) Celsius to Fahrenheit conversion



**Ex. No.:3**

**PAYROLL PROCESSING**

**Aim:**

To develop a Java application with employee class and generate pay slips for the employees with their gross and net salary.

**Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

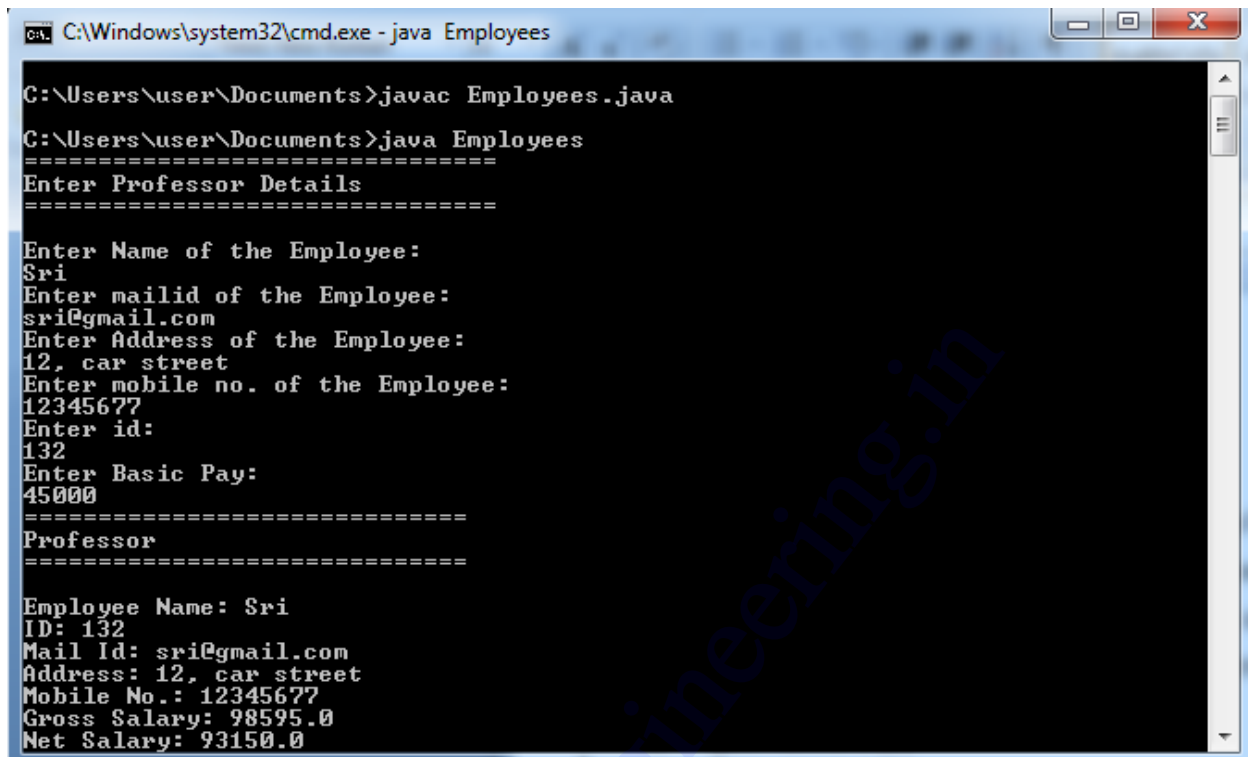
**Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

**Procedure:**

1. Start the program
2. Create Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members.
3. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class.
4. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund.
5. Generate pay slips for the employees with their gross and net salary.
6. Stop the program

### Sample output:



```
C:\Windows\system32\cmd.exe - java Employees

C:\Users\user\Documents>javac Employees.java

C:\Users\user\Documents>java Employees
=====
Enter Professor Details
=====
Enter Name of the Employee:
Sri
Enter mailid of the Employee:
sri@gmail.com
Enter Address of the Employee:
12, car street
Enter mobile no. of the Employee:
12345677
Enter id:
132
Enter Basic Pay:
45000
=====
Professor
=====

Employee Name: Sri
ID: 132
Mail Id: sri@gmail.com
Address: 12, car street
Mobile No.: 12345677
Gross Salary: 98595.0
Net Salary: 93150.0
```

### Result:

Thus the Java application has been created with with employee class and pay slips are generated for the employees with their gross and net salary.

### Outcome:

Thus the course outcome (CO1) has been attained by applying the concept of inheritance to generate Payroll Processing application using Java.

### Applications:

- (1) EB Bill Generation
- (2) Income Tax Calculation

## Ex. No.:4

### ADT STACK

#### Aim

To design a Java interface for ADT Stack using array.

#### Hardware Requirements:

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

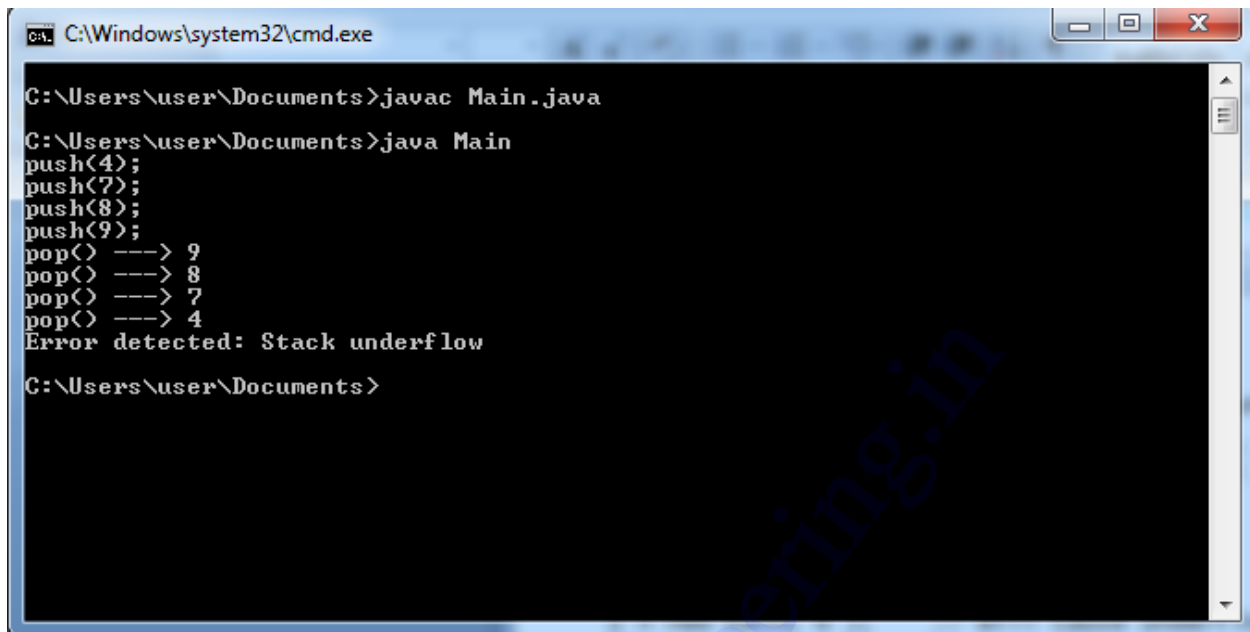
#### Software Requirements:

- Ubuntu 14.04 operating system
- JDK 1.8

#### Procedure:

1. Start the program
2. Define the interface.
3. Read the elements using array.
4. Initialize stackTop pointer as zero,
5. Define and use the method Push() to insert the elements into the stack with 'STACK OVERFLOW' condition.
6. Define and use the method pop() to remove an element from an array with 'STACK UNDERFLOW' condition
7. Display the output.

**Sample output:**



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

C:\Users\user\Documents>javac Main.java

C:\Users\user\Documents>java Main
push(4);
push(7);
push(8);
push(9);
pop() ----> 9
pop() ----> 8
pop() ----> 7
pop() ----> 4
Error detected: Stack underflow

C:\Users\user\Documents>
```

**Result:**

Thus the design and implementation of ADT Stack using array has successfully executed.

**Outcome:**

Thus the course outcome (CO2) has been attained by applying the concept of array to generate ADT Stack using Java.

**Applications:**

- (1) ADT Queue
- (2) ADT circular queue

**Ex. No.:5**

## **STRING OPERATIONS**

### **Aim**

To write a program to perform string operations using ArrayList.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

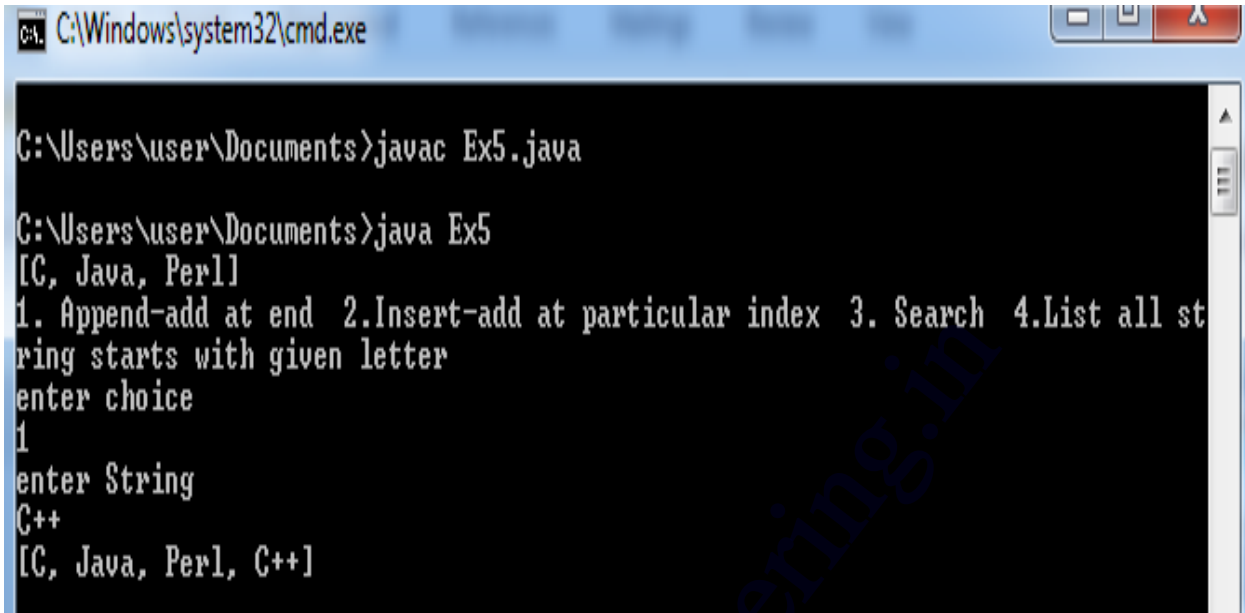
### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

### **Procedure:**

1. Start the program
2. Add the String as an object to List.
3. Get the choice from the user and do according to the choice
  - a. Append-add at end
  - b. Insert-add at particular index
  - c. Search
  - d. List all string starts with given letter.
3. Display the result
4. Stop the program.

**Sample output:**



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

C:\Users\user\Documents>javac Ex5.java

C:\Users\user\Documents>java Ex5
[C, Java, Perl]
1. Append-add at end  2.Insert-add at particular index  3. Search  4.List all st
ring starts with given letter
enter choice
1
enter String
C++
[C, Java, Perl, C++]
```

**Result:**

Thus the implementation of string operations using array list has been successfully executed.

**Outcome:**

Thus the course outcome (CO2) has been attained by applying the concept of arraylist for string manipulations using Java.

**Applications:**

- (1) Reverse of the String.
- (2) String Matching.
- (3) Counting the number of vowels and consonants in a line.

## Ex. No.:6

### ABSTRACT CLASS

#### Aim

To write a Java Program to create an abstract class named Shape and provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape.

#### Hardware Requirements:

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

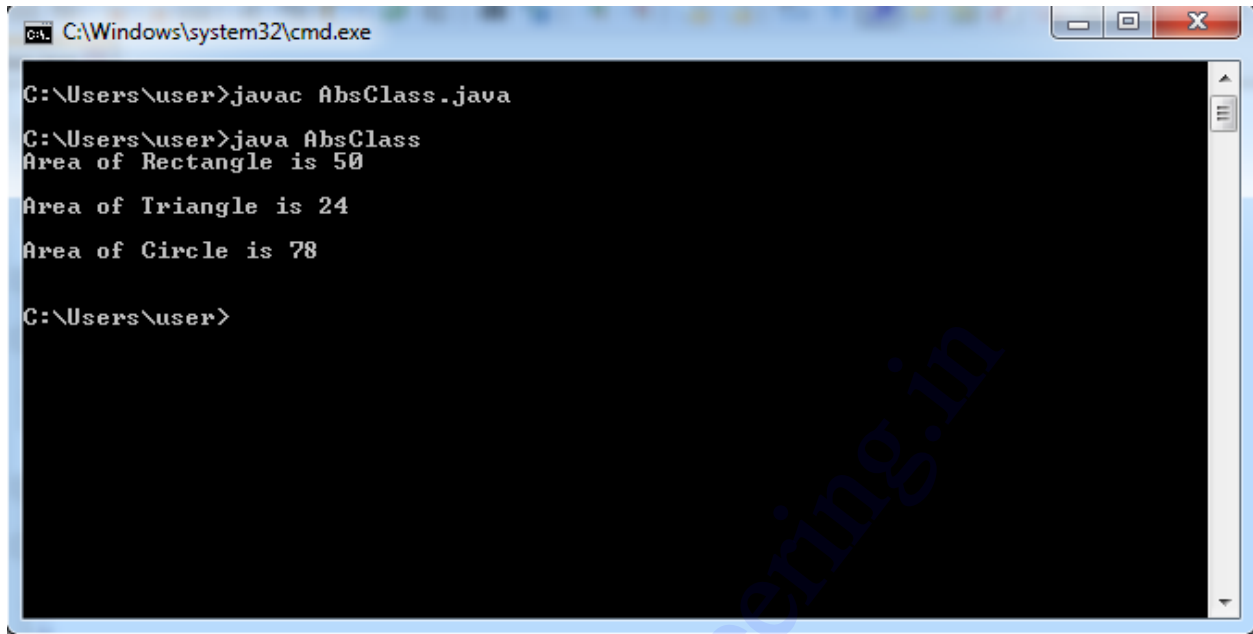
#### Software Requirements:

- Ubuntu 14.04 operating system
- JDK 1.8

#### Procedure:

1. Start the program
2. Define the abstract class **shape**.
3. Define the class Rectangle with PrintArea() method that extends(makes use of) Shape.
4. Define the class Triangle with PrintArea() method that extends(makes use of) Shape.
5. Define the class Circle with PrintArea() method that extends(makes use of) Shape.
6. Print the area of the Rectangle, Triangle and Circle .
7. Stop the Program.

**Sample output:**



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

C:\Users\user>javac AbsClass.java
C:\Users\user>java AbsClass
Area of Rectangle is 50
Area of Triangle is 24
Area of Circle is 78

C:\Users\user>
```

**Result:**

Thus the design and implementation of Abstract class has been successfully executed.

**Outcome:**

Thus the course outcome (CO2) has been attained by applying the concept of abstract class using Java.

**Applications:**

- (1) Volume of the Cube
- (2) Length of the Cube
- (3) Area of Triangle



**Ex. No.:7**

## **EXCEPTION HANDLING**

### **Aim**

To write a Java program to implement user defined exception handling.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

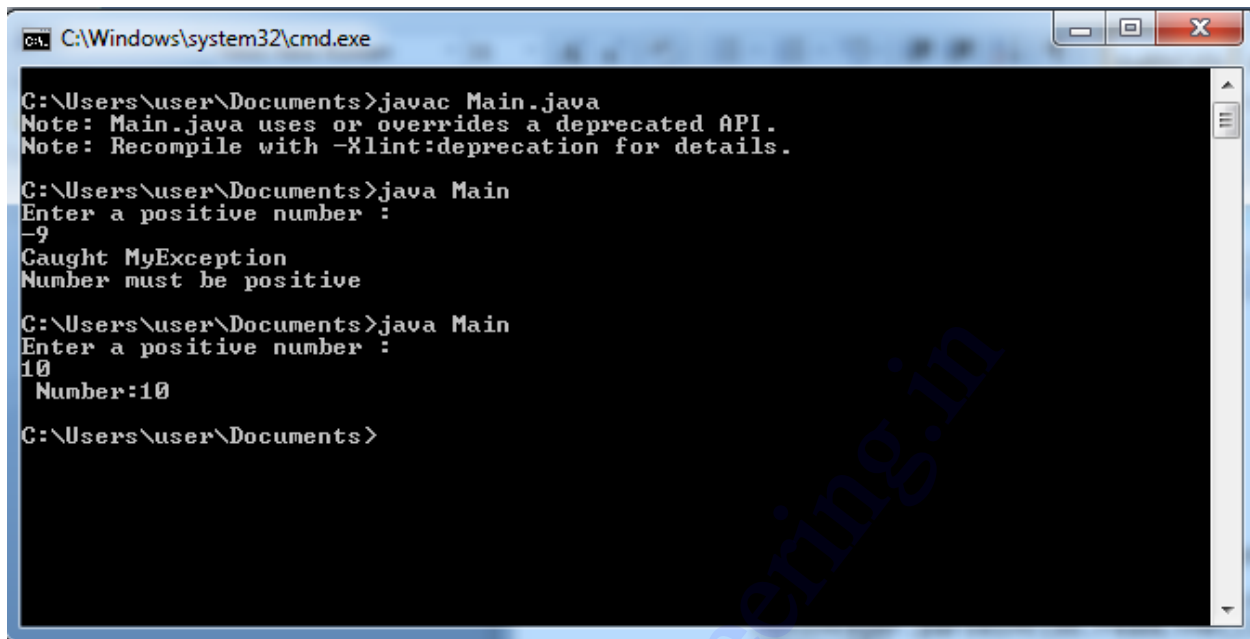
### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

### **Procedure:**

1. Start the program
2. Define the exception for getting a number from the user.
3. If the number is positive print the number as such.
4. If the number is negative throw the exception to the user as '**Number must be positive**'.
5. Stop the Program.

### Sample output:



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

C:\Users\user\Documents>javac Main.java
Note: Main.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\user\Documents>java Main
Enter a positive number :
-9
Caught MyException
Number must be positive

C:\Users\user\Documents>java Main
Enter a positive number :
10
Number:10

C:\Users\user\Documents>
```

### Result:

Thus the user defined exception has been successfully implemented.

### Outcome:

Thus the course outcome (CO2) has been attained by applying the concept of Exception handling for an user defined exception using Java.

### Applications:

- (1) Throwing exception for Checking the @ symbol in Email Id
- (2) Throwing exception for password mismatch.

**Ex. No.: 8**

## **FILE INFORMATION**

### **Aim**

To write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

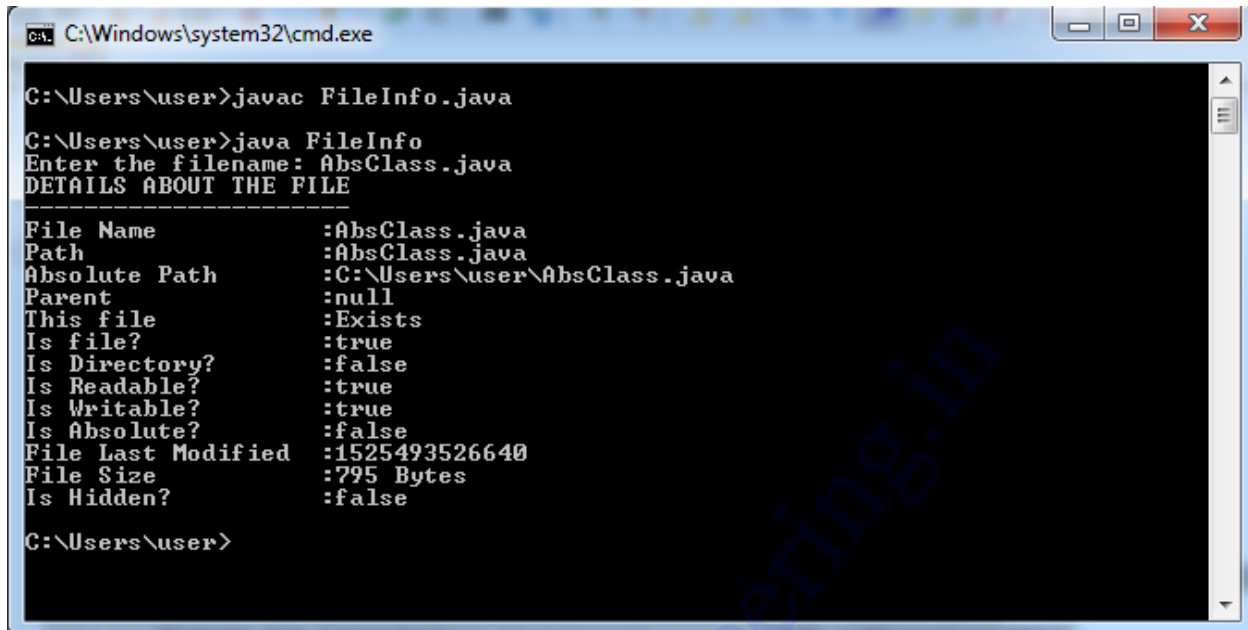
### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

### **Procedure:**

1. Start the program
2. Read the filename from the user.
3. Use getName() Method to display the filename.
4. Use getPath() Method to display the path of the file.
5. Use getParent() Method to display its parent's information.
6. Use exists() Method to display whether the file exist or not
7. Use isFile() and isDirectory() Methods to display whether the file is file or directory.
8. Use canRead() and canWrite() methods to display whether the file is readable or writable.
9. Use lastModified() Method to display the modified information.
10. Use length() method to display the size of the file.
11. Use isHidden() Method to display whether the file is hidden or not.

### Sample output:



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

C:\Users\user>javac FileInfo.java

C:\Users\user>java FileInfo
Enter the filename: AbsClass.java
DETAILS ABOUT THE FILE
-----
File Name      :AbsClass.java
Path           :AbsClass.java
Absolute Path  :C:\Users\user\AbsClass.java
Parent        :null
This file      :Exists
Is file?       :true
Is Directory?  :false
Is Readable?   :true
Is Writable?    :true
Is Absolute?   :false
File Last Modified :1525493526640
File Size      :795 Bytes
Is Hidden?     :false

C:\Users\user>
```

### Result:

Thus the information of the file has been displayed successfully using various file methods.

### Outcome:

Thus the course outcome (CO1) has been attained y file operations using Java..

### Applications:

- (1) IRCTC chart display (RAC reservation)
- (2) A2B menu display

**Ex. No.: 9**

## **MULTITHREADING**

### **Aim**

To write a java program that implements a multi-threaded application.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

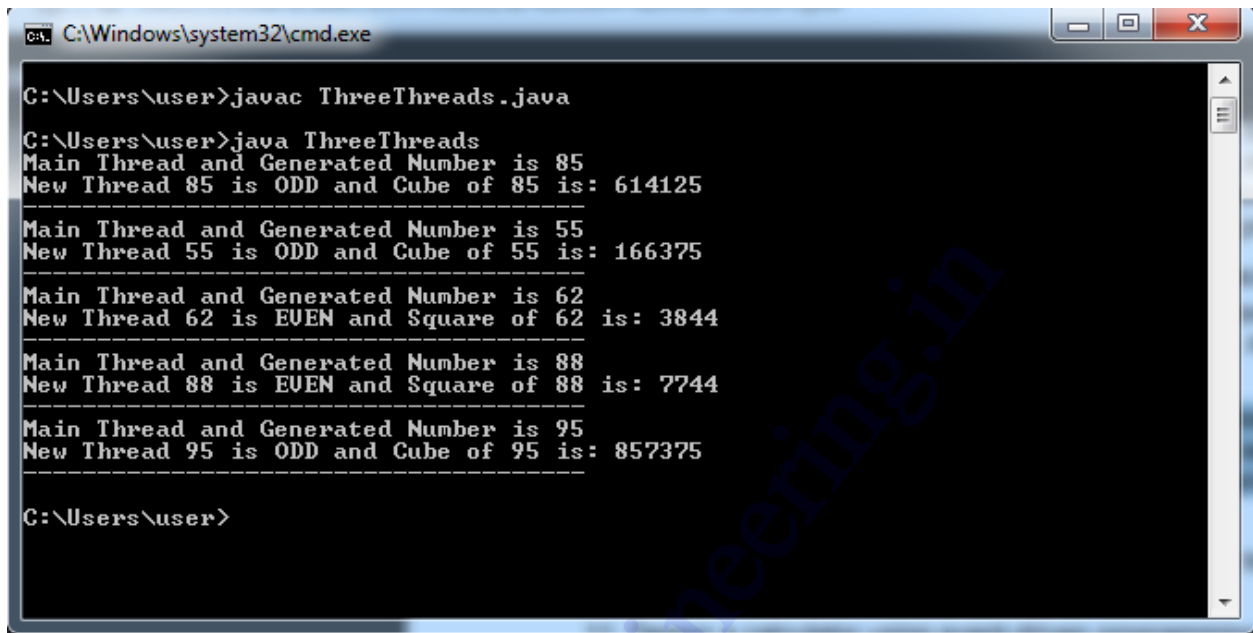
### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

### **Procedure:**

1. Start the program
2. Design the first thread that generates a random integer for every 1 second .
3. If the first thread value is even, design the second thread as the square of the number and then print it.
4. If the first thread value is odd, then third thread will print the value of cube of the number.
5. Stop the program.

**Sample output:**



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

C:\Users\user>javac ThreeThreads.java
C:\Users\user>java ThreeThreads
Main Thread and Generated Number is 85
New Thread 85 is ODD and Cube of 85 is: 614125
-----
Main Thread and Generated Number is 55
New Thread 55 is ODD and Cube of 55 is: 166375
-----
Main Thread and Generated Number is 62
New Thread 62 is EVEN and Square of 62 is: 3844
-----
Main Thread and Generated Number is 88
New Thread 88 is EVEN and Square of 88 is: 7744
-----
Main Thread and Generated Number is 95
New Thread 95 is ODD and Cube of 95 is: 857375
-----
C:\Users\user>
```

**Result:**

Thus the implementation of multithreading has been done using three threads.

**Outcome:**

Thus the course outcome (CO2) has been attained by applying the concept of multithreading to generate odd numbers and its square using Java.

**Applications :**

- (1) Multiplication Table Printing (3,5,8<sup>th</sup> table)
- (2) Printing Area of the cube, square and circle

**Ex. No.: 10**

## **GENERIC FUNCTION**

### **Aim**

To write a java program to find the maximum value from the given type of elements using a generic function.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

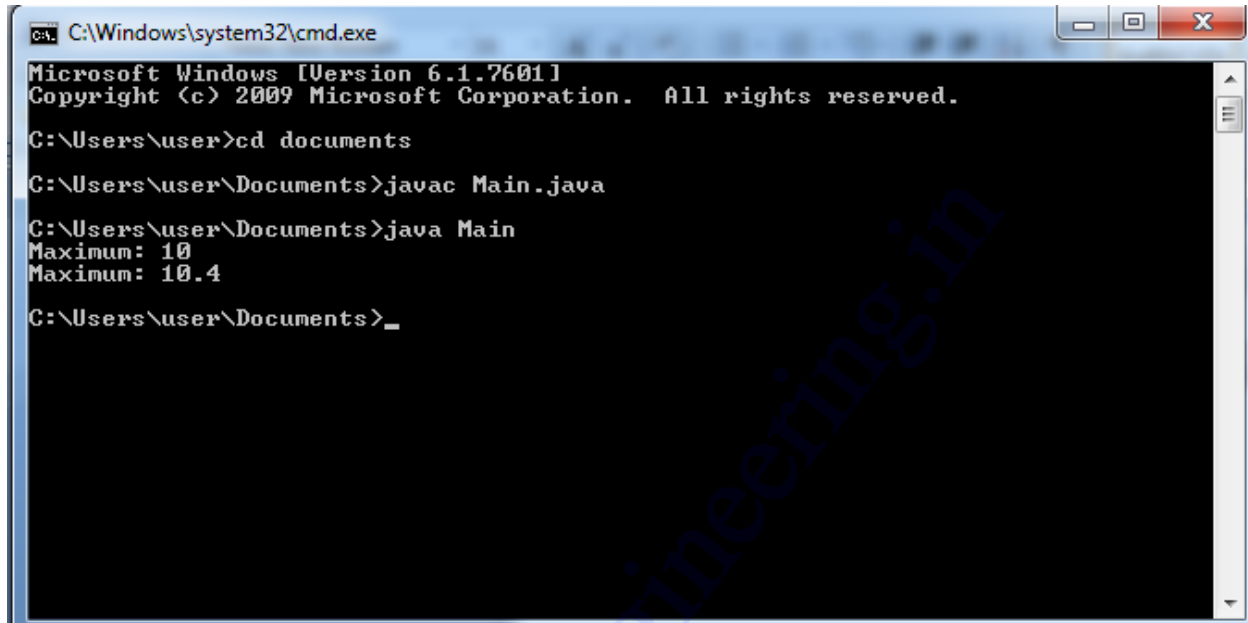
### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

### **Procedure:**

1. Start the program
2. Define the array with the elements
3. Sets the first value in the array as the current maximum
4. Find the maximum value by comparing each elements of the array
5. Display the maximum value
6. Stop the program.

**Sample output:**



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>cd documents
C:\Users\user\Documents>javac Main.java
C:\Users\user\Documents>java Main
Maximum: 10
Maximum: 10.4
C:\Users\user\Documents>_
```

**Result:**

Thus the implementation of generic function is achieved for finding the maximum value from the given type of elements.

**Outcome:**

Thus the course outcome (CO3) has been attained by applying the concept of generic function to generate maximum value and minimum value. .

**Applications:**

- (1) Finding even values
- (2). Finding the sum of values



**Ex. No.: 11**

## **CALCULATOR**

### **Aim**

To design a calculator using event-driven programming paradigm of Java for Decimal manipulations and Scientific manipulations.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8

### **Procedure:**

1. Start the program
2. Using the swing components design the buttons of the calculator
3. Use key events and key listener to listen the events of the calculator.
4. Do the necessary manipulations.
5. Stop the program.

### Sample Output:

```
C:\Windows\system32\cmd.exe - java DecimalCalculator

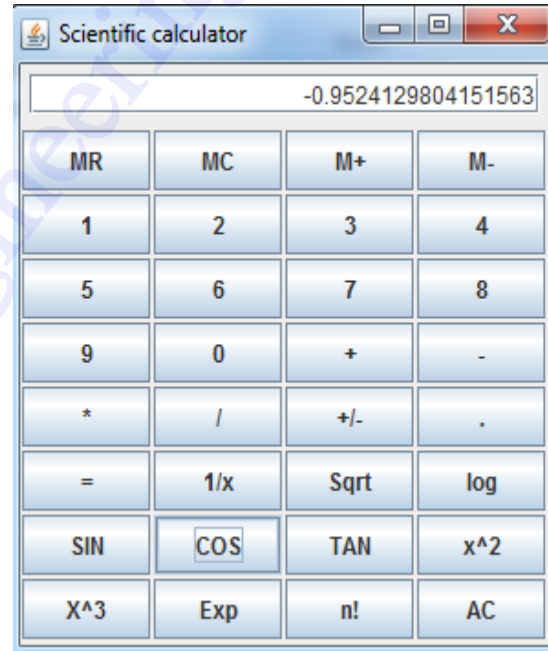
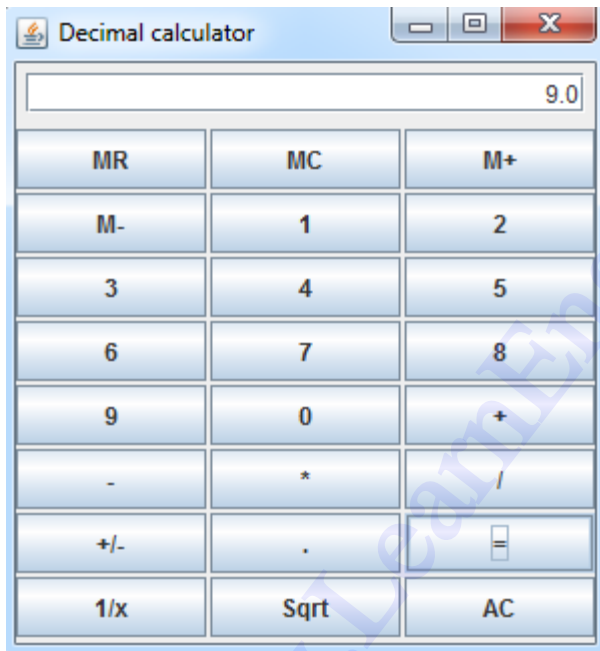
C:\Users\user>javac DecimalCalculator.java

C:\Users\user>java DecimalCalculator
```

```
C:\Windows\system32\cmd.exe - java ScientificCalculator

C:\Users\user>javac ScientificCalculator.java

C:\Users\user>java ScientificCalculator
```



### Result:

Thus the implementation of generic function is achieved for finding the maximum value from the given type of elements.

### Outcome:

Thus the course outcome (CO3) has been attained by applying the concept of event handling to design calculator using Java.

**Applications:**

(1) Age calculator

(2) EB calculator

[www.LearnEngineering.in](http://www.LearnEngineering.in)

**Ex. No.: 12**

## **VEHICLE MANAGEMENT SYSTEM**

### **Aim**

To design a vehicle management system using Java.

### **Hardware Requirements:**

- Pentium IV with 2 GB RAM
- 160 GB HARD Disk
- Monitor 1024 x 768 colour

### **Software Requirements:**

- Ubuntu 14.04 operating system
- JDK 1.8
- Notepad
- Internet Explorer 4.0

### **Procedure:**

1. Start the program
2. Using the swing components design the necessary layout
3. Add the details using JDBC(Java Data Base Connectivity).
4. Get the details of vehicle number, make, Fuel type , Vehicle type ,Insurer ,Rupees per kilometer , Cost and company from the user
5. Store it in the database and do the necessary manipulations.
6. Stop the program

Sample output:

### Admin Page



### VEHICLE MANAGEMENT – ADMIN

Vehicle Details :



Result:

Thus the design of vehicle management system using Java has been successfully executed.

Outcome:

Thus the course outcome (CO1) has been attained by generating a real world application using Java.

## MINIPROJECTS

1. Airline Reservation System
2. Mark sheet Preparation system
3. NAAC online application creation
4. Library Management System
5. Converting RGB image to Gray Image
6. Health Care System
7. App development
8. Income Tax System
9. Vehicle Tracking System
10. Ebanking System