



**NEW HORIZON  
COLLEGE OF ENGINEERING**

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC  
Accredited by NAAC with 'A' Grade, Accredited by NBA

## **E BANKING SERVICE**

**A MINI PROJECT  
REPORT**

*Submitted by*

**DIVYASHREE S**

*In partial fulfillment for the award of*

*the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## Certificate

*This is to certify that the mini project work titled*

### E BANKING SERVICE

*Submitted in partial fulfillment of the degree of Bachelor of Engineering*

**DIVYASHREE S**

**1NH18CS061**

*DURING*

*EVEN SEMESTER 2019-2020*

*For*

**19CSE48**

Signature of Reviewer

Signature of HOD

### SEMESTER END EXAMINATION

*Name of the Examiner*

*Signature with date*

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

## E BANKING SERVICE

### ORIGINALITY REPORT

7%

SIMILARITY INDEX

%

INTERNET SOURCES

7%

PUBLICATIONS

%

STUDENT PAPERS

### PRIMARY SOURCES

- 1

" Internet banking in India The Reserve Bank of India constituted a working group on Internet Banking. The group divided the internet banking products in India into 3 types based on the levels of access granted. ", University/Business and Administrative studies/Management Studies, 2011-07-16

2%

Publication
- 2

Kishori Sharan. "Beginning Java 8 Fundamentals", Springer Science and Business Media LLC, 2014

2%

Publication
- 3

S G Ganesh, Tushar Sharma. "Oracle Certified Professional Java SE 7 Programmer Exams 1Z0-804 and 1Z0-805", Springer Science and Business Media LLC, 2013

1%

Publication
- 4

Singh, Shamsheer, and Naveen J. Sirohi. "Internet banking services as tool of CRM: a study of customer satisfaction in the National Capital Region, Delhi", International Journal of

1%

Electronic Customer Relationship Management,  
2014.

Publication

- 
- |                            |   |                |
|----------------------------|---|----------------|
| <b>5</b>                   | "Inheritance and its type in Object Oriented Programming using C++", International Journal of Recent Technology and Engineering, 2019 | <b>&lt;1 %</b> |
| <small>Publication</small> |   |                |
- 
- |                            |   |                |
|----------------------------|---|----------------|
| <b>6</b>                   | Ivor Horton. "Beginning C++", Springer Science and Business Media LLC, 2014 | <b>&lt;1 %</b> |
| <small>Publication</small> |   |                |
- 
- |                            |   |                |
|----------------------------|---|----------------|
| <b>7</b>                   | Ivor Horton. "Ivor Horton's Beginning ANSI C++: The Complete Language", Springer Nature, 2004 | <b>&lt;1 %</b> |
| <small>Publication</small> |   |                |
- 
- |                            |   |                |
|----------------------------|---|----------------|
| <b>8</b>                   | "Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops", Springer Science and Business Media LLC, 2010 | <b>&lt;1 %</b> |
| <small>Publication</small> |   |                |
- 
- |                            |  |                |
|----------------------------|--|----------------|
| <b>9</b>                   | Ivor Horton. "More about Program Structure", Wiley, 2012 | <b>&lt;1 %</b> |
| <small>Publication</small> |  |                |
- 
- |                            |  |                |
|----------------------------|--|----------------|
| <b>10</b>                  | Juan Manuel Durán. "Chapter 2 Units of Analysis I: Models and Computer Simulations", Springer Science and Business Media LLC, 2018 | <b>&lt;1 %</b> |
| <small>Publication</small> |  |                |
-

11

Josh Juneau. "Java 9 Recipes", Springer  
Science and Business Media LLC, 2017

Publication

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On

## ABSTRACT

The main aim of this project is to create an e banking application which provides a user-friendly interface to the customers performing transactions. The application is developed to save the time of customer as well as the bank. It also provides easy and secured facilities to the customers.

In this project the user can perform operations such as checking balance, depositing amount, withdrawing amount, checking previous transaction and payment of loan can be calculated. The user can perform these operations just by sitting in house in a few clicks instead of physically going to the bank. This saves a lot of time and

The programming language used to build this application is java. The object-oriented concepts used in this project are class, objects and inheritance. Using object-oriented concepts, the application can be developed easily and efficiently.

By using this application, the drawbacks from the existing system can be overcome. Hence the new proposed system brings lot of advantages.

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha**, Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr. Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Mrs. Tinu**, Asst professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

**DIVYASHREE S (1NH18CS061)**

## TABLE OF CONTENTS

|  |    |
|--|----|
| 1. INTRODUCTION.....                   | 4  |
| 1.1 PROBLEM DEFINITION                 |    |
| 1.2 OBJECTIVES                         |    |
| 1.3 EXPECTED OUTCOMES                  |    |
| 1.4 HARDWARE AND SOFTWARE REQUIREMENTS |    |
| 2. OBJECT ORIENTED CONCEPTS.....       | 6  |
| 2.1 CLASSES                            |    |
| 2.2 OBJECTS                            |    |
| 2.3 INHERITANCE                        |    |
| 2.4 POLYMORPHISM                       |    |
| 2.5 ABSTRACT CLASS                     |    |
| 2.6 MULTITHREADING                     |    |
| 2.7 I/O FUNCTIONS                      |    |
| 2.8 JAVA PACKAGES                      |    |
| 2.9 EXCEPTION HANDLING                 |    |
| 3. ALGORITHM.....                      | 19 |
| 4. IMPLEMENTATION.....                 | 21 |
| 5. RESULTS.....                        | 24 |
| 6. CONCLUSION.....                     | 28 |



## **LIST OF FIGURES**

| <b><u>Fig. no</u></b> | <b><u>Figure description</u></b> | <b><u>page no.</u></b> |
|-----------------------|----------------------------------|------------------------|
| 2.1                   | class example                    | 7                      |
| 2.31                  | single inheritance               | 9                      |
| 2.32                  | multiple inheritance             | 9                      |
| 2.33                  | multilevel inheritance           | 10                     |
| 2.34                  | hierarchical inheritance         | 10                     |
| 2.35                  | hybrid inheritance               | 11                     |
| 2.5                   | Life Cycle of Thread             | 14                     |
| 2.7                   | java packages                    | 16                     |
| 2.8                   | Exception hierarchy              | 17                     |

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 PROBLEM DEFINITION**

The existing banking management system has many drawbacks as every task is taken care by human beings. The time taken is more as all the calculations are done manually and requires more man power. There is less security and all the documents must be taken care from disasters. The complexity increases when many customers enter the bank at the same time hence it is time consuming.

#### **1.2 OBJECTIVES**

After understanding the existing system, the main objective of this project is to provide easy and secure facilities to the customers. By using E banking services, the customer gets lots of benefits like: 1. The Customer need not have to wait in long queues in bank. Hence the customer saves a lot of time. 2. The customer can perform banking operations just by sitting in house over internet and few clicks on gadgets. The proposed system not only saves time but it also provides data security and uses less man power, reduces the human interaction with customer for simple doubts / issues and consumption of resources like paper and pens are comparatively less. Henceforth the investment made on this system helps on longer run and the investment made on integrating e-banking services is less comparatively.

#### **1.3 EXPECTED OUTCOMES**

The expected outcomes are: -

- The customer will be able to deposit money into his account.
- The customer will be able to withdraw money from his account.
- The customer can check balance after performing operations like deposit and withdrawal of cash.

- The customer can view the previous transaction done by the customer in his account.
- The customer will be able to check the monthly and annual payment to be paid for the loan amount taken.

### 1.4 HARDWARE AND SOFTWARE REQUIREMENTS

|                  |                                   |
|------------------|-----------------------------------|
| Processor        | : Any processor above 500MHz      |
| RAM              | : 512Mb                           |
| Hard disk        | : 10 GB                           |
| Input device     | : standard keyboard and mouse     |
| Output device    | : VGA and high resolution monitor |
| Operating system | : windows XP                      |
| Software         | : eclipse IDE                     |
| Front end        | : Java                            |

## CHAPTER 2

### OBJECT ORIENTED CONCEPTS

Object oriented programming is a concept of using objects which consists data in the form of fields and procedures. Most widely used programming languages such as C++, java, python etc support object-oriented programming.

The advantages of using object-oriented programming are: -

- They are easy to understand.
- They have clear modular structure for programs.
- They can reused in other programs.
- It reduces the development cost.
- Large programs can be written with a minimum if object oriented concepts are used.
- It also enhances the program modularity.

The object-oriented concepts are: -

1. Class
2. Objects
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation
7. Association
8. Aggregation
9. Composition

#### 2.1 CLASS

A class is a group of similar entities. It is a logical component and not a physical entity.

It is an entity which describes how an object will behave.

A class declaration consists of the following: -

1. Modifiers
2. Class name
3. Keywords
4. The class body within curly braces

For example, consider expensive cars such as Rolls Royce, BMW, Ferrari, Lamborghini etc here 'expensive cars' is a class and every single physical car is an object of the class 'expensive cars'.

Another example of a class is shown below

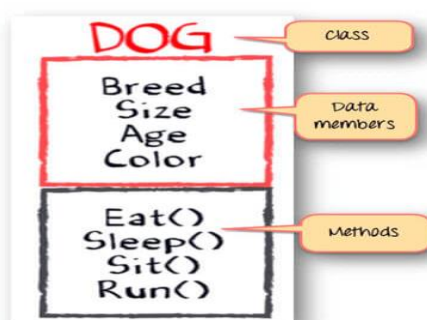


Fig 2.1 class example

To create a class, the keyword class is used.

Syntax:

Class < class name>

```
{
field;
Method;
}
```

## 2.2 OBJECTS

An entity that has state and behaviour is known as object.

In this project when we consider objects it can have three characteristics.

1. State: It represents the value of the object.
2. Behaviour: It represents the functionality of the object such as deposit, withdrawal etc.

3. Identity: An object identity is usually implemented via a singular ID. The ID value is not visible to the external users. However, its used internally by the JVM to spot each object uniquely.

Every real-world object contains state and behaviour.

For example, pen is an object then its colour will be its state. It is used to write therefore its behaviour is writing.

Syntax: -

Declaring objects:

Class name object name;

Constructing objects:

Object name = new class name (parameters);

Calling object methods:

Return type = object name. method name (parameters);

## 2.3 INHERITANCE

Inheritance consists of two categories

1. Sub class – It is also known as child class. It inherits the properties from other class.
2. Super class – It is also known as base class. It is the class that is being inherited from.

The concept of inheriting the properties and functionalities of super class to sub class is known as inheritance. The main advantage of inheritance is the reusability of the code.

To inherit the properties to sub class the keyword 'extends' is used.

The various types of inheritance are

### 1. Single inheritance

A sub class inheriting the properties and functionalities from a single super class is known as single inheritance

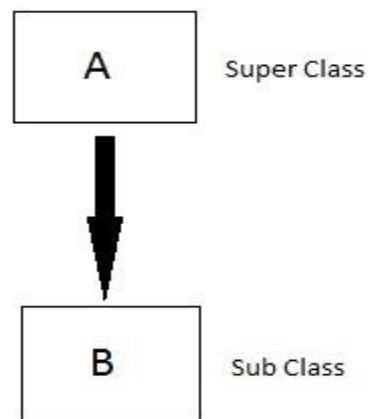


Fig 2.31 Single Inheritance

## 2. Multiple inheritance

A sub class inheriting the properties and functionalities from more than one super class is known as multiple inheritance.

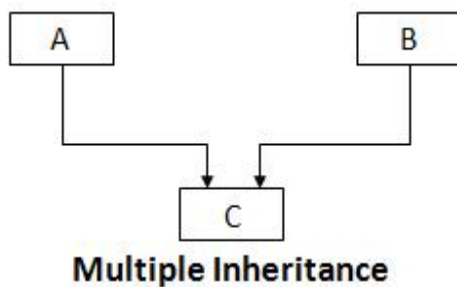


Fig 2.32 Multiple Inheritance

## 3. Multi-level inheritance

Considering there are three classes A, B and C. Class B inheriting the properties from class A and class C inheriting the properties from class B is known as multi-level inheritance.

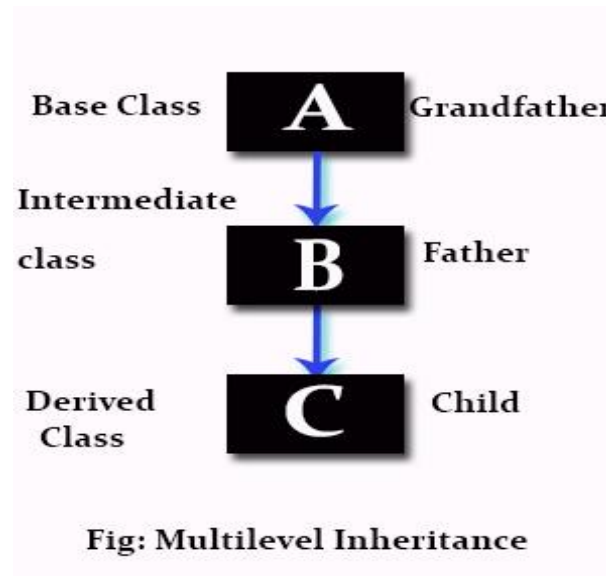


Fig 2.33 Multilevel Inheritance

#### 4. Hierarchical inheritance

The process of more than one sub class inheriting the properties from the same super class is known as hierarchical inheritance.

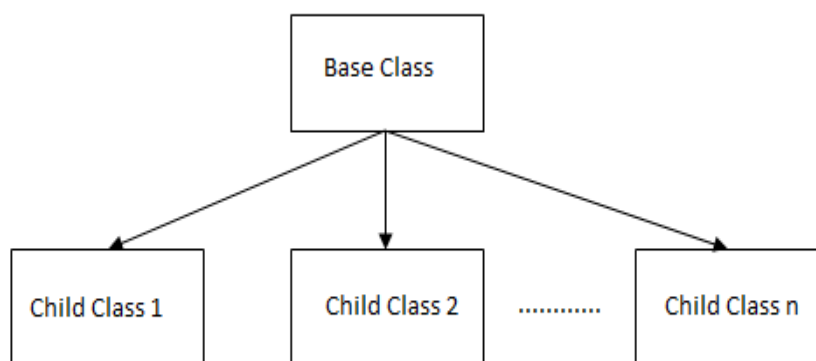


Fig: Hierarchical Inheritance

Fig 2.34 Hierarchical Inheritance

#### 5. Hybrid Inheritance

The combination of single and multiple inheritance is known as hybrid inheritance.



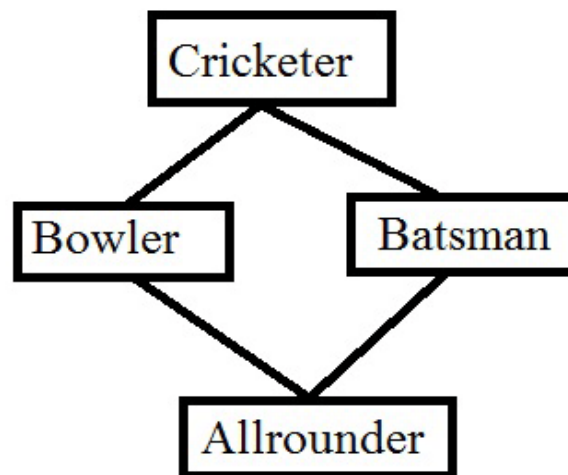


Fig 2.35 Hybrid Inheritance

## 2.4 POLYMORPHISM

Polymorphism is a feature of opps which allows us to perform single task in many ways. It is the capability of a method to do different things based on the object that it is acting upon.

There are two types of polymorphism

### 1. Static polymorphism

Polymorphism which is resolved during compiler time is known as static or compile time polymorphism. An example of static polymorphism is method overloading.

### 2. Dynamic polymorphism

Dynamic polymorphism is also known as dynamic method dispatch. It is a process in which the call to an overridden method is resolved during the runtime. Hence it is known as dynamic polymorphism or runtime polymorphism.

## 2.5 ABSTRACT CLASS

The process of hiding the unnecessary details and highlighting the necessary details is known as abstraction.

A class which has the abstract keyword is known as abstract class.

An abstract class may or may not contain abstract methods but if the class has an abstract method then it must be declared abstract.

It cannot be instantiated if the class is declared abstract.

To use an abstract class, it must be inherited from other class and provide implementations to the abstract method in it.

Uses of abstract class: -

Let us consider we have a class called animal and a method sound () and sub classes of it like cat, dog, lion, pig etc. Since the sound of the animal differs from one animal to other animal we cannot implement this in the super class because every sub class will have to override it as lion may say “roar” and a dog may say “woof”. Therefore, we know that every animal child classes will override and there is no use of implementing this method in parent class. Hence making this an abstract class will be the ideal option as we force all the sub classes to implement this method and it’s not necessary to give implementation to this method in the super class.

## 2.6 MULTITHREADING

The smallest unit of processing is known as single thread. It is used by using a thread class. There are two types of thread,

1. User thread
2. Daemon thread

The applications are

- Maintenance is low
- It reduces overhead in the application as single thread execute in the system

There is a process in JAVA, in which you can execute two or more threads at the same time (simultaneously) for the maximum utilization of CPU. Applications with MULTITHREADED option allows executing two or more threads run concurrently. so we call it to be as CONCURRENCY in JAVA. In this process, each thread runs parallel to each other. It does not

allocate separate memory area; hence the memory is saved. The time taken in the switching of context between the threads is also less.

Advantages of MULTITHREAD:

1. Since the threads are independent, users are not blocked and enables us to perform multiple operations at the same time.
2. Also, other threads won't get affected if one thread meets an exception because the threads are independent.

Different stages of life cycle of thread are shown in the diagram below:

- a. NEW
- b. RUNNABLE
- c. RUNNING
- d. WAITING
- e. DEAD

1. NEW: the first phase of the cycle, we create the thread by using "THREAD CLASS". The thread will remain in the same state until the program STARTS. It is called as BORN THREAD

2. RUNNABLE: In this Phase, With the START METHOD the thread is invoked at the instance. To finish the execution the control of thread is given to scheduler. Whether to run the thread is depended on the scheduler.

3. RUNNING: This is the phase then when the threads start the execution, the state is changed to "RUNNING" state. One thread is selected by the scheduler from thread pool, and the execution is started in the application.

4. WAITING: Since multiple threads are running in the application, in this stage the thread as to wait. Synchronization is required in between the threads. Hence, till the other thread is executed, one thread has to wait. So, we refer this as WAITING stage.

5. DEAD: The thread is terminated in this stage, after the complete processing of the thread from running state it is in "DEAD STATE"

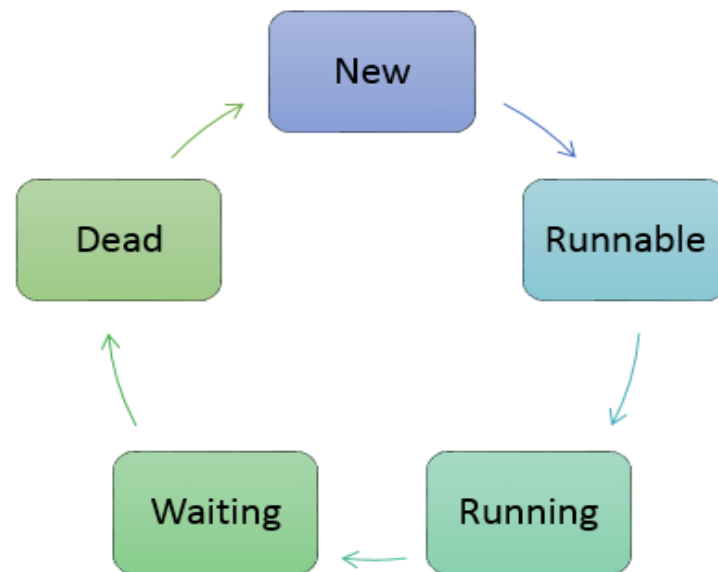


Fig 2.5 Life cycle of Thread

## 2.7 I/O FUNCTIONS

I/O functions are used to process the input and produce the output.

The concept of stream is used to make the I/O operations fast

The three standard/default streams are

1. System.in: -

This standard input stream is used to read characters from standard input devices like keyboard etc.

2. System. Out: -

This is the standard output stream which is used to produce the output in the output device.

The various print functions used in output statements are

- Print()
- Println()
- Printf()

### 3. System. Err: -

It is the standard error stream which is used to display all the error the program might show on the computer screen.

The various print functions used are

- Print()
- Println()
- Printf()

## 2.8 JAVA PACKAGES

A group of similar type of classes, interfaces and sub packages is known as java package.

There are two types of packages: -

### 1. Built in packages

The various built in packages are java, Lang, awt, javax, swing, Io, net, util, sql etc.

### 2. User defined packages

The package created by the user is called user defined package.

Advantages of using a package: -

- It is used to categorize the interfaces and classes so that they are easily maintained.
- It removes naming collision.
- It provides access projection.

The keyword package is used every time to create a new package .

Syntax: -

Package <package name>;

Example: package new pack;

Public class simple;

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

{

    System.out.println ("welcome to package");

}

}
```

The different ways of importing a package are

- Import package.\*;
- Import package.classname;
- Fully qualified name.

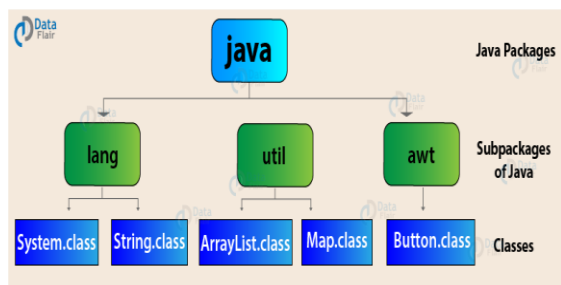


Fig 2.7 Java packages

## 2.9 EXCEPTION HANDLING

An unwanted or unexpected event which occurs during the execution of the program (during run time) and stops the normal flow of instructions is called an exception.

Exception handling is a technique of handling the run time errors such as IO Exception, class not found Exception, SQL Exception etc.

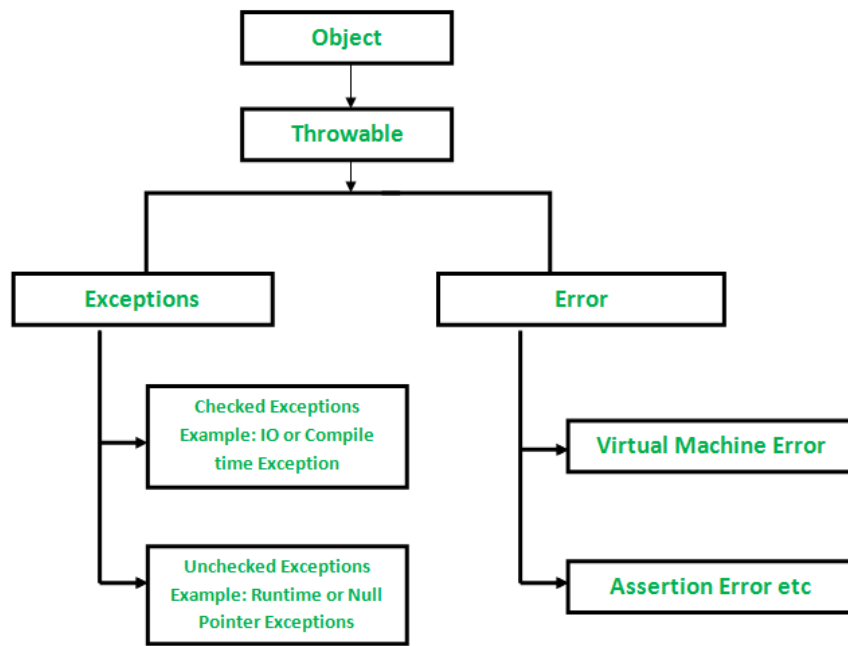


Fig 2.8 Exception Hierarchy

The above diagram shows the hierarchy of exception.

There are two types of exception in java

1. checked exception.

The errors which occur during the run time is known as checked exceptions.

2. Unchecked exception.

The errors which occur during the run time is known as unchecked exceptions.

In java five keywords are used to handle exceptions

The five keywords are

1. Try
2. Catch
3. Finally
4. Throw
5. Throws

- Try: - The program code which might throw an exception is put within the try block. A try block must always be followed by either catch or finally.
- Catch: - The catch block is used to handles the exceptions occurred in the try block. A catch block must always be used after try block.

- Finally: -Finally block is used to execute important code such as closing connection, stream etc. It is always executed whether an exception is handled or not.
- Throw: - Keyword throw is used to throw an exception.
- Throws: - The keyword throws is used to declare the exception and not throw the exception. It provides information to the programmer that an exception may occur in a specific method.



## CHAPTER 3

### ALGORITHM

- Create a package for banking application and in the package create a new class (super class) and declare the variable datatypes used in the program. Create a constructor and pass the values of customer name and customer id.
- Create a method called interest which calculates the monthly and total payment of the loan to be paid.
  - Declare the value of the interest rate
  - Monthly interest rate is calculated by dividing the interest rate by 1200
  - The user must enter the no. of years and the loan payment
  - Monthly payment is calculated by the formula  $\text{loan amount} \times \text{monthly interest rate} / (1 - 1/\text{Math. POW}(1 + \text{monthly interest rate}, \text{number of years} \times 12))$ .
  - Total payment is calculated by the formula  $\text{monthly payment} \times \text{number of years} \times 12$ .
  - Print the monthly and total payment
- Create another class in the package banking application (sub class) which inherits from the super class and define the methods deposit, withdraw and previous transaction.
  - Deposit

When the amount is not equal to zero, add the amount to balance and update the previous transaction with the amount entered.
  - Withdraw

When the amount is not equal to zero, subtract the amount from the balance and update the previous transaction with the amount deducted.

➤ Previous transaction

If the previous transaction is greater than zero, print the deposited amount. If the previous transaction is less than zero, print the withdrawn amount. If both the cases are false then print no transaction occurred.

- Open a new project, click on windows builder then click swing designer and select application window.
- When the program opens click on design and create a login page using the Labels, password field and button which consists of the username, password, login and reset.
- In the coding part of reset button set text to null. For the login button declare the username and password and give an if condition which is if the username and password match then the banking code must open other else print a message stating invalid details.
- Import the banking application package into the login systems so that the banking operations can be performed by the customer when they login.
- Create a main class, instantiate an object for the sub class and call the switch case using the object.

## CHAPTER 4

### IMPLEMENTATION

A banking application can be created in java using object-oriented programming. In this project the super class contains the method print which is used to display the customer details and a method interest which calculates the loan amount. The subclass consists of the methods deposit, withdraw and previous transaction. The methods print and interest are inherited from the super class.

#### MODULE1

In the method interest, the loan amount to be payed is calculated. The customer is asked to enter the year and the amount they would like to take as loan. The below code is used to calculate the monthly and yearly payment of loan.

```
Void interest ()
{
    Scanner ob = new Scanner (System.in);
    System.out.println ("The interest rate provided is 7.5");
    Double monthlyInterestRate = 7.5/1200;
    System.out.println ("Enter the number of years");
    Int numberOfYears = ob.nextInt ();
    System.out.println ("enter the loan amount");
    Double loanAmount = ob.nextDouble ();
    Double monthlyPayment= loanAmount*monthlyInterestRate/ (1-1/Math.
    POW (1+monthlyInterestRate, numberOfYears*12));
    Double total Payment = monthlyPayment* numberOfYears*12;
    System.out.println ("The monthly payment is Rs "+(int)
    (monthlyPayment*100/100.0));
    System.out.println ("The total Payment is Rs "+(int)(totalPayment*100/100.0));
}
```

#### MODULE 2

Deposit method is used to deposit money into the customer's account.

The below code is used to deposit the money.

```
Void deposit(int amount)
{
```

```
    if(amount!=0)
    {
        balance=balance+amount;
        previous1=amount;
    }
}
```

### MODULE 3

Withdraw method is used to withdraw money from the existing bank account.

The below code is used for the withdrawal of money.

```
void withdraw(int amount)
{
    if(amount!=0)
    {
        balance=balance-amount;
        previous1=-amount;
    }
}
```

### MODULE 4

In previous transaction method the customer will be able to view the last transaction which has occurred in his/her account.

The below code is used to check the previous transaction.

```
void previoustransaction()
{
    if(previous1>0)
    {
        System.out.println("Deposited : "+previous1);
    }
    else if(previous1<0)
    {
        System.out.println("Withdrawn : "+ Math.abs(previous1));
    }
    else
    {
        System.out.println("no transaction occurred");
    }
}
```

## MODULE 5

To create a login page: -

```
if(Password.contains("pass") && Username.contains("user"))
{
    txtPassword.setText(null);
    txtUsername.setText(null);

    Previous info = new Previous();
    Previous.main(null);

}
else
{
    JOptionPane.showMessageDialog(null,"Invalid Details","Login Error!",
    JOptionPane.ERROR_MESSAGE );
    txtPassword.setText(null);
    txtUsername.setText(null);
}
```

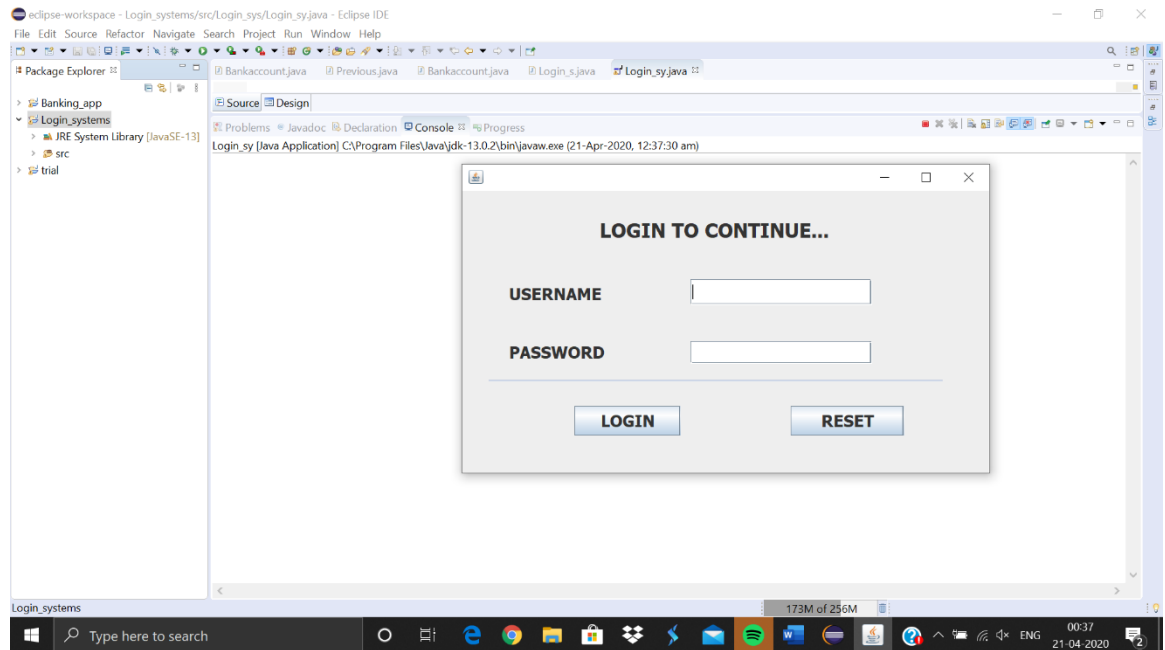
For reset button: -

```
public void actionPerformed(ActionEvent e)
{
    txtUsername.setText(null);
    txtPassword.setText(null);
}
```

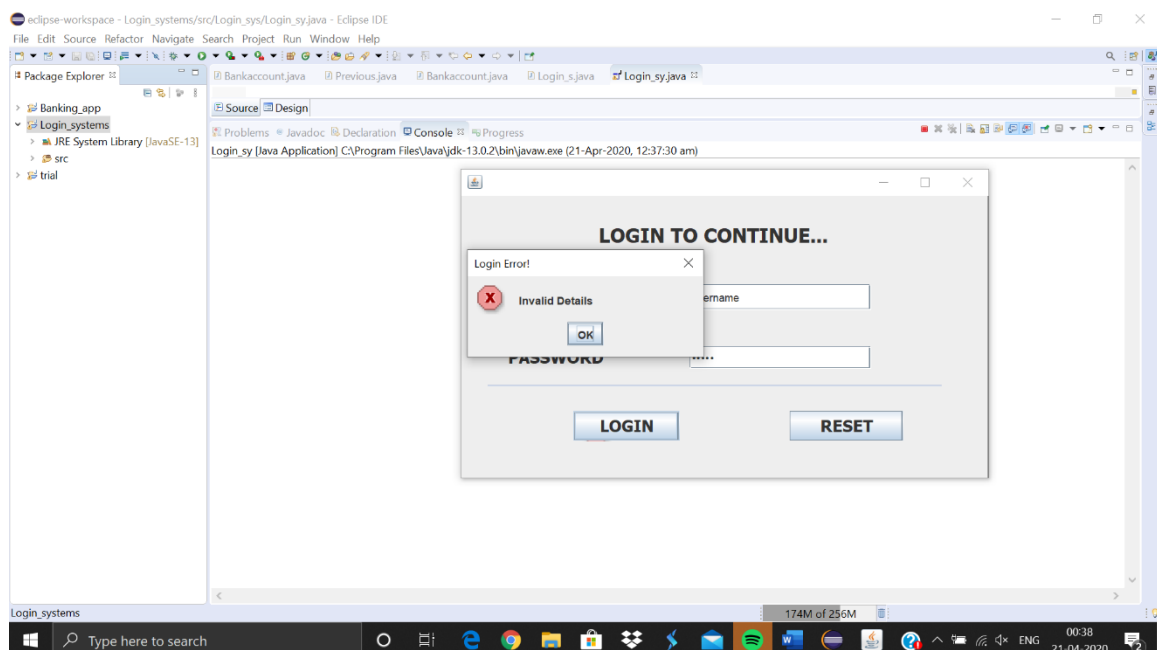
## CHAPTER 5

## RESULTS

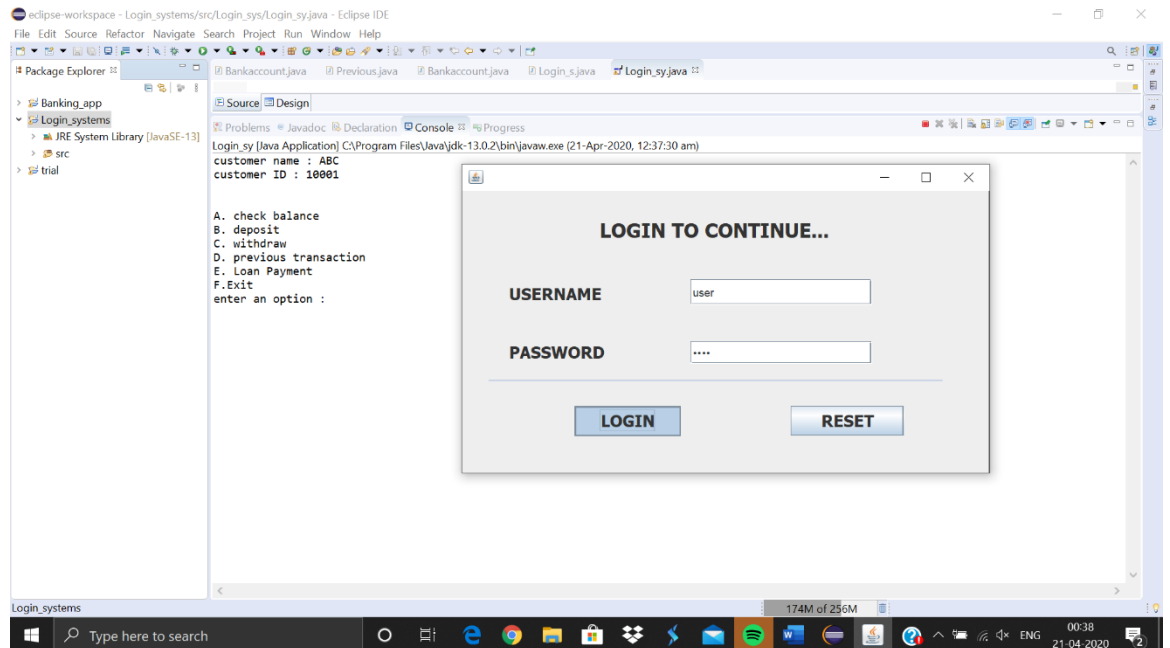
### ➤ Login page



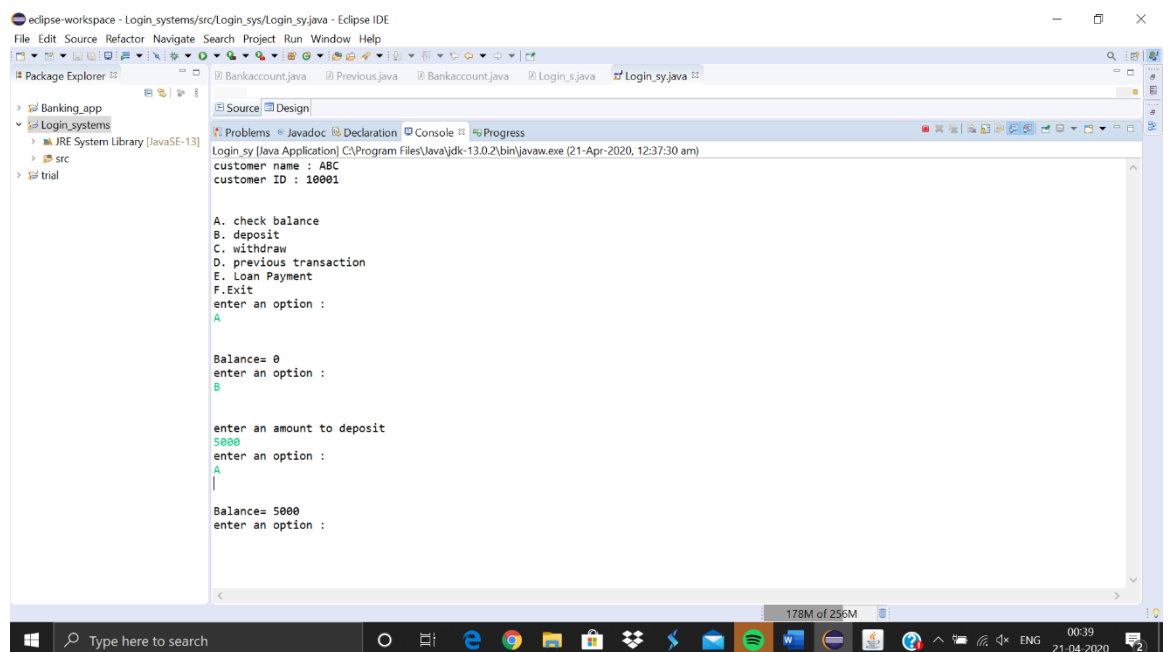
### ➤ Login page when incorrect details are entered



### ➤ Login page with correct details



### ➤ Depositing money

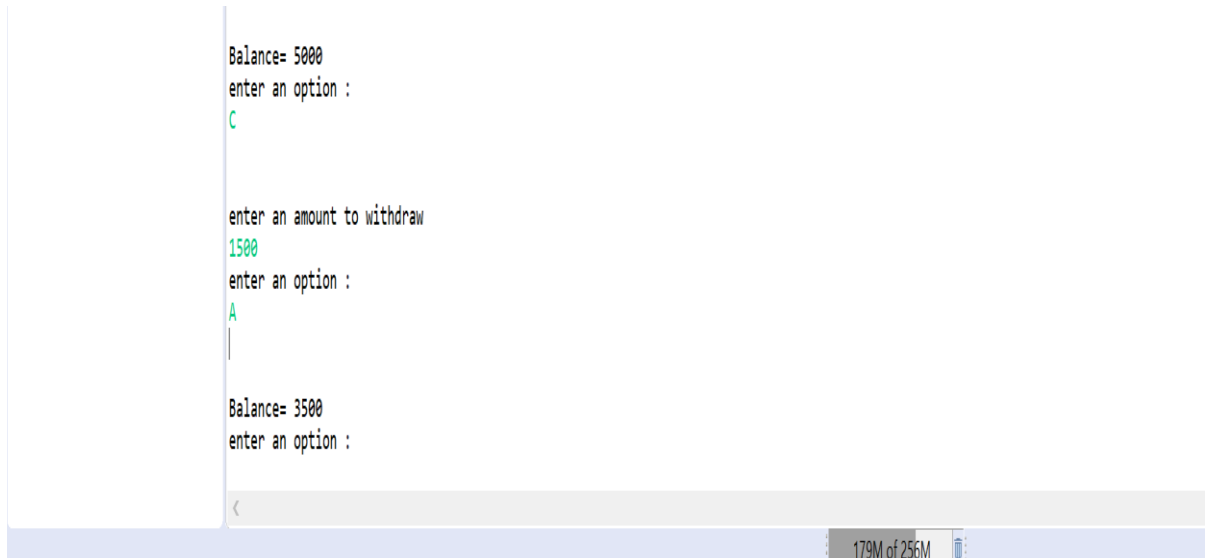


### ➤ Withdrawal of money

```
Balance= 5000
enter an option :
C

enter an amount to withdraw
1500
enter an option :
A
|

Balance= 3500
enter an option :
```

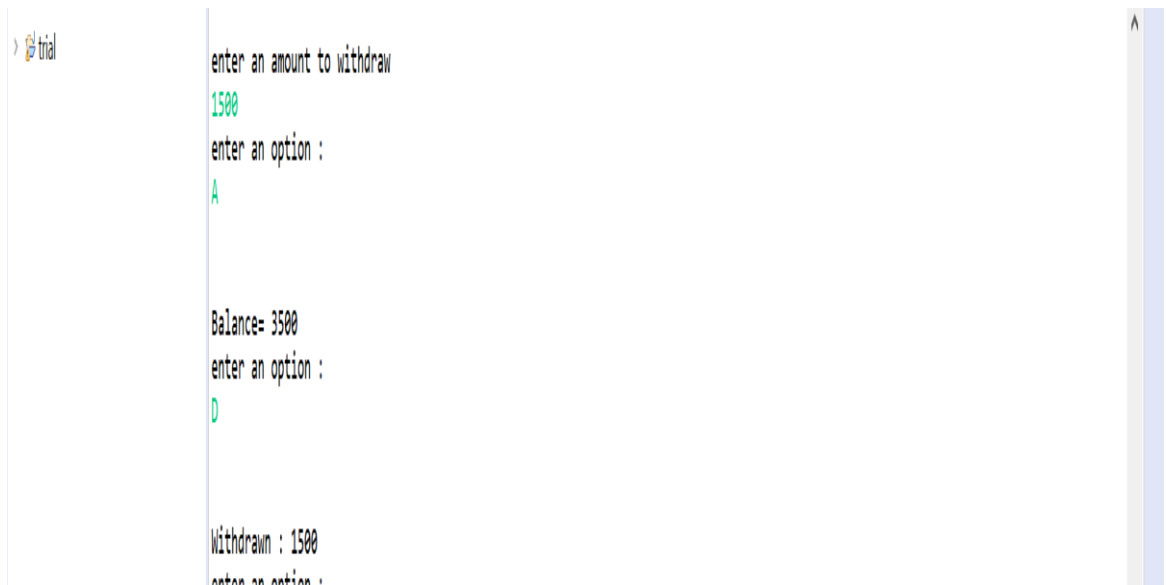


### ➤ Checking previous transaction

```
> trial
enter an amount to withdraw
1500
enter an option :
A

Balance= 3500
enter an option :
D

Withdrawn : 1500
enter an option :
```





### ➤ Loan payment

```
enter an option :
E

The interest rate provided is 7.5
Enter the number of years
3
enter the loan amount
200000
The monthly payment is Rs 6221
The total Payment is Rs 223964
enter an option :
F

.....

Thank you for using our services
```

## CHAPTER 6

### CONCLUTION

Internet banking is changing the banking industry and is having major effects on banking relationships. The net banking, thus, "now is more of a norm rather than an exception in many developed countries" due to the fact that it is the economical way of providing banking services. Banking is now no longer confined to the traditional brick and mortar branches, where one has to be at the branch in person, to withdraw cash or deposit a cheque or request a statement of accounts. Providing internet banking is increasingly becoming a need to have than a nice to have services.

By using the object-oriented concepts, it is easy to develop a banking application which can perform operations like deposit, withdraw, checking balance, checking previous transaction and checking the loan payment.

We finally conclude that by using this project we can provide a user-friendly interface between the bank and the customer.

## REFERENCE

<https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>

[https://www.tutorialspoint.com/java/java\\_inheritance.htm](https://www.tutorialspoint.com/java/java_inheritance.htm)

An Introduction To Object-Oriented Programming With Java by C. Thomas Wu