**BANK SYSTEM**

**Introduction:**

Why I am doing this project?

Answer: This project is based on different types of uses in Java program for example : Abstraction , Exception handling etc. and learn how the bank system works. To gain the idea of OOP1 (JAVA) and to work on different concept of java, analysis the problems and develop myself I’m doing this project.

**Problem Analysis:**

What kind of problem is being solved with this project? Who will use this application?

Answer: The project is about bank system so this application can be useable for the bank and the people having bank account. The main aim of this project is to develop a system for bank account so that it can be use easily and quickly, which is not possible with the manuals for example open an account, deposit money and withdraw at any time. So those things are solved by the project.

The programming language Java is used for this application and different concepts of java is being used in this project.

**Logical Analysis:**

How much logic has been applied in this application? Are the applied logics working properly? Is there any logical limitations?

(No more than 100 words)

Answer: The logics used in this project are:

1. The class Bank which extends Bank System (An abstract class) used to write methods of :

1. openAccount : to open several accounts by user( using Scanner)

2. showAccount : to show the name , balance etc.

3. fordeposit : by addition [long Data type used as it can store 64-bit]

4. forwithdraw : by subtraction.

5. search : Boolean search is used to show account if the user input result is true

2. ExBank which extends Bank is used for operating the system. Switch-case is used to operate the methods if the user inputs are matched. The loop will be continuing until exit (5) is pressed. Here the process shows the main menu for user help so that it can be use easily.

3. MainBank: is the main of the code and there object is created and the methods call by ob.

So, these are used in the project which works properly and there have no logical limitations on this project.

**OOP Concept Analysis:**

What are the OOP1 principles used in this project? How have been they applied? Explain the OOP concepts used in this application (No more than 3 sentences per concept)?

(No more than 160 words)

Answer: Class: Collection of objects is called class. It is a logical entity which is used in this project.

Abstraction: Hiding internal details and showing functionality. Abstract class and interface is used to achieve abstraction .In this project abstract methods are:

abstract void deposit();

abstract void withdraw();

abstract void interest();

In this code there have abstract methods and no abstract body.

Polymorphism: method overriding and overloading used to achieve. BUT

In this code, overriding is used for abstract method.

Exception: Exception concept is used (try-throw-catch) in some cases to handle from occurring error.

**Impact of this Project:**

What impact this project will have on the society and economy? How will people be benefitted from your project?

Answer: As the main aim of this project is to develop a system for bank account it has been developed to carry out the processes easily and quickly, which is not possible with the manuals. So from this project people can easily co-operate with Banks as it will be easy to withdraw or deposit money at any time and it is secure to save money though bank.

Therefore, there is no chance of theft here. So people will be benefitted from this project.

**Limitations and Possible Future Improvements:**

Answer: There are some limitations on that project.

1. In case of interest there have some issues because interest can be applied on saving accounts and as this project is not showing anything about saving accounts so there have some limitations for that case.

2. User has to open account before all the process of main menu. User can not open an account after main menu’s process starts.

Improvements:

There is no use of graphical interface so the project can be improved by using graphical interface.