**Banking Record System (report)**

**Enrolment number – E22CSEU1292**

**Name – Advik Tushir**

**Assumption:** To effectively display interest change, we are finding interest based on a per-day concept. Instead of making plans over months or years, we shorten the scale of finding interest to days. We exclusively deal with the Savings and Current types of accounts.

**Approach:** To keep a record of the customer, we used file handling. To save data, we use csv structured files. To get the required result, we used the manipulator function and string. In the bank, we applied the notion of interest calculation. To obtain the current time from the system, we use functions.

**Header Files Involved:**

**#include <iostream>** -> The preset library function, often known as header files, is what is utilised for input and output. All programme functions, including cout, cin, and others, are contained in the header file called iostream.

**#include <fstream>** -> File streams include two member functions specifically designed to read and write binary data sequentially: write and read . The first one ( write ) is a member function of ostream (inherited by ofstream ). And read is a member function of istream (inherited by ifstream ). Objects of class fstream have both.

**#include <cstring>** -> This header file defines several functions to manipulate C strings and arrays.

**#include <iomanip>** -> The header <iomanip> is part of the Input/output library of the C++ Standard Library. It defines the manipulator functions resetiosflags() , setiosflags() , setbase() , setfill() , setprecision() , and setw() .

**#include <cstdio>** -> Input and Output operations can also be performed in C++ using the C Standard Input and Output Library. This library uses what are called streams to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system

**#include <conio.h>** -> conio.h is a C header file used mostly by MS-DOS compilers to provide console input/output.

**#include <ctime>**  -> The ctime() function in C++ converts the given time since epoch to a calendar local time and then to a character representation. A call to ctime(time) is a combination of asctime() and localtime() functions, as asctime(localtime(time)) .

**#include <stdlib.h>** -> stdlib.h is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others.

**Using namespace std:**

A namespace is a form of scope in C++ that holds its own definitions for variables, functions, etc. For example, both cout and cin , along with some useful tokens like endl , are defined inside of std for use.

**Global Variable :**

int main\_exit; -> variable used globally so that can be used for accepting a value for exit or main menu or for previous menu;

**Global Function:**

Functions created globally so that can be accessed on any function.

**void fordelay(int);**  -> A function in which there is a loop which runs 100000000 times, to make some delay.

**void close(void);** -> A function to display a message on closing the program.

**int main();** -> It involves the option for employee and customer.

**void menu\_e();** -> It provide menu for employee, to perform few operations.

**void employee();** -> It provide access to the employee menu after inputting correct Employee Id and Password.

**void customer();**  -> It provide access to the customer menu after inputting correct Employee Id and Password for the existing user and direct access to the new user**.**

**void menu\_c();**  -> It provide menu for customer, to perform few operations

**Class :**

A class named Bank which have few data member and few member functions.

**Data members involved:**

**string account\_number;** -> To store account no. of the Customer

**string name;** -> To store name of the Customer

**string dob;** -> To store date of birth of the Customer

**string age;** -> To store age of the Customer

**string date;** -> To store only date of creation of account of the Customer

**string address;** -> To store address of the Customer

**string phone;** -> To store phone no. of the Customer

**string depo;** -> To store balance of the Customer

**string acc\_type;** -> To store account type of the Customer

**string password;** -> To store password of the account of the Customer

**string interest;** -> To store interest of the Customer

**Data Functions involved:**

**void read\_data();** -> Functions to read data of the Customer

**void acc\_write();** -> Functions to write data of the Customer in csv formatted file

**void modify\_account();** -> Functions to modify data of the Customer

**void search\_rec();** -> Functions to search data of the Customer

**void deposit\_withdraw();** -> Functions to deposit and withdraw of the Customer

**void display\_all();** -> Functions to display all account holders details of the Customer

**void transfer();** -> Functions to transfer amount from a account no. to another of the Customer

**Functions Used:**

**getline( )** -> To read a data either from a file or from user through keyboard

**cin.ignore( )** -> To encounter the buffer error

**fflush(stdin)** -> To encounter buffer error

**atoi( )** ->To Convert string type numerical value to integer type value

**to\_string( )** ->To Convert a numeric value to a string type value

**obj.good( )** ->It goes to the end of the file

**obj.eof( )** -> It goes to the end of the file

**system(“cls “)** -> To clear the screen

**system(“color 5”)** -> To change the color of the text display on the screen

**obj.open( )** -> To open a file in several mode.

**obj.close( )** -> To close a file

**seekp( )** ->It goes to the mentioned position when open in write mode

**seekg( )** -> It goes to the mentioned position when open in read mode

**tellp( )** ->It tells the position of the cursor when open in write mode

**tellg( )** -> It tells the position of the cursor when open in read mode

**.c\_str( )** ->To Convert a string into Character array

**time\_t now = time(0); struct tm \*ltm = localtime(&now);** -> To get the current date and time from the system

**clear( )**  ->To change the position of the cursor to the beginning

**remove( )** -> To Remove a file from system

**rename( )** -> To Rename a file in the system