**DATA STRUCTURES AND ALGORITHMS**

**CODE FOR BANK MANGEMENT SYSTEM**

**CODE**

|  |
| --- |
| /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // BANK MANAGEMENT SYSTEM //  // DATA STRUCTURES AND ALGORITHMS //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  #include <iostream>  #include <fstream>  #include <windows.h>  #include<conio.h>  #include <string.h>  using namespace std;  char level=220;  // CREATING CLASS NODE //  class Node {  public:  int user\_Id;  string Name;  string city;  string Cnic;  string mobile\_Number;  int amount;  Node\* next;  Node\* t;  };  // Stores the head of the Linked List  Node\* head = new Node();  // Check Function to check that if Record Already Exist or Not  bool check(int x)  {  // Base Case  if (head == NULL)  return false;  Node\* t = new Node;  t = head;  // Traverse the Linked List  while (t != NULL) {  if (t->user\_Id == x)  return true;  t = t->next;  }  return false;  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // INSERT RECORD //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // Function to insert the record  void Insert\_Record(int user\_Id, string Name,  string city, string Cnic,string mobile\_Number,int amount)  {  int i,n;  if (check(user\_Id)) {    return;  }  // Create new Node to Insert Record  Node\* t = new Node();  t->user\_Id =user\_Id;  t->Name = Name;  t->city = city;  t->Cnic = Cnic;  t->amount= amount;  t->mobile\_Number=mobile\_Number;  t->next = NULL;  // Insert at Begin  if (head == NULL|| (head->user\_Id >= t->user\_Id)) {  t->next = head;  head = t;  }  // Insert at middle or End  else {  Node\* c = head;  while (c->next != NULL && c->next->user\_Id < t->user\_Id) {  c = c->next;  }  t->next = c->next;  c->next = t;  }  cout << "\t\t\t\t RECORD INSERTED SUCCESSFULLY\n";  system("cls");  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // DEPOSIT MONEY //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void deposit\_money(int a) //Deposit amount  {  Node \*t=head;  int b;  b=a;  t->amount=t->amount+b;  t=t->next;  }    /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // WITHDRAW MONEY //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void withdrawal(int a) //Withdraw amount  {  Node \*t=head;  int b;  b=a;  t->amount=t->amount-b;  t=t->next;  }    /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // MODIFY ACCOUNT //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void modify(){  Node \*t=head;  ofstream outfile;  outfile.open("bank.txt", ios::out | ios::app |ios::in);  cout<<"\n\t\t>>>>>ACCOUNT NO>>>>>>>"<<t->user\_Id<<endl;  cout <<"\t\t ENTER NAME OF USER:: ";  cin >>t->Name;  outfile<<t->Name;  cout <<"\t\t ENTER USER ID:: ";  cin >>t->user\_Id;  outfile<<t->user\_Id;  cout <<"\t\t ENTER CITY:: ";  cin >>t->city;  outfile<<t->city;  cout <<"\t\t ENTER CNIC:: ";  cin >>t->Cnic;  outfile<<t->Cnic;  cout<<"\t\t ENTER AMOUNT:: ";  cin>>t->amount;  outfile<<t->amount;  cout<<"\t\t ENTER MOBILE NUMBER:: ";  cin>>t->mobile\_Number;  outfile<<t->mobile\_Number;  Insert\_Record(t->user\_Id, t->Name, t->city, t->Cnic,t->mobile\_Number,t->amount);  outfile.close();  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // SEARCH RECORD //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void Search\_Record(int user\_Id)  {  // if head is NULL  system("cls");  int i;  if (!head) {  cout << "\t\t\t\t NO SUCH RECORD AVAILABLE\n";  return;  }  // Otherwise  else {  Node\* p = head;  while (p) {  if (p->user\_Id == user\_Id) {  ifstream infile;  infile.open("bank.txt",ios::in|ios::out);  cout<<"==========================================================================================================="<<endl;  cout<<" ACCOUNTS DETAILS "<<endl;  cout<<"==========================================================================================================="<<endl;  cout<<"Account No. Name City CNIC Mobile No Balance"<<endl;  cout<<"==========================================================================================================="<<endl;  infile>>user\_Id>>p->Name>>p->city>>p->Cnic>>p->mobile\_Number>>p->amount;  cout<<p->user\_Id <<"\t\t"<< p->Name<<"\t\t"<< p->city <<"\t\t"<< p->Cnic<<"\t\t"<<p->mobile\_Number<<"\t\t"<<p->amount<<endl;  infile.close();  return;  }  p = p->next;  }  if (p == NULL)  cout << "\t\t\t\t NO SUCH RECORD AVAILABLE\n";  }  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // DELETE RECORD //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  int Delete\_Record(int user\_Id)  {  system("cls");  Node\* t = head;  Node\* p = NULL;  // Deletion at Begin  if (t != NULL && t->user\_Id == user\_Id) {  head = t->next;  delete t;  cout << "\t\t\t\t RECORD DELETED SUCCESSFULLY\n";  return 0;  }  // Deletion Other than Begin  while (t != NULL && t->user\_Id != user\_Id) {  p = t;  t = t->next;  }  if (t == NULL) {  cout << "\t\t\t\t RECORD DOES NOT EXIST\n";  return -1;  p->next = t->next;  delete t;  cout << "\n\t\t\t\t RECORD DELETED SUCCESSFULLY\n";  return 0;  }  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // SHOW USER RECORD //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void Show\_Record(int user\_Id)  {  system("cls");  Node\* p = head;  if (p == NULL) {  cout << "\t\t\t\t NO RECORD AVAILABLE\n";  }  else {    while (p != NULL) {  ifstream infile;  infile.open("bank.txt",ios::in|ios::out);  cout<<"==========================================================================================================="<<endl;  cout<<" ACCOUNTS DETAILS "<<endl;  cout<<"==========================================================================================================="<<endl;  cout<<"Account Number Name City CNIC Mobile No Balance"<<endl;  cout<<"==========================================================================================================="<<endl;  infile>>user\_Id>>p->Name>>p->city>>p->Cnic>>p->mobile\_Number>>p->amount;  cout<<p->user\_Id <<"\t\t"<< p->Name<<"\t\t"<< p->city <<"\t\t"<< p->Cnic<<"\t\t"<<p->mobile\_Number<<"\t\t"<<p->amount<<endl;  p = p->next;  infile.close();  }  }  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // MODIFY RECORD //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void modify\_account(int num){  bool found=false;  Node \*t=head;;  ofstream outFile;  outFile.open("bank.txt",ios::binary|ios::in|ios::out);  if(!outFile)  {  cout<<"\t\t\t\tFile could not be open !! Press any Key..."<<endl;  return;  }  if(t->user\_Id==num)  {  cout<<"\t\t\t\t>>>>PREVIOUS RECORD>>>>>"<<endl;  Show\_Record(t->user\_Id);  cout<<"\t\t\t\t>>>>PREVIOUS RECORD>>>>>"<<endl;  getch();  cout<<"\n\n\t\t\t\tEnter The New Details of account"<<endl;  modify();  cout<<"\n\n\t\t\t\t Record Updated";  found=true;  }  outFile.close();  if(found==false)  cout<<"\n\n\t\t\t\t Record Not Found ";  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // DEPOSIT WITHDRAW FUNCTION //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  void deposit\_withdraw(int num,int choice)  {  Node \*t=head;  int amnt;  int a;  a=choice;  if(a==1)  {  fstream infile("bank.txt", ios::binary|ios::in|ios::out);  infile>>t->user\_Id>>t->Name>>t->city>>t->Cnic>>t->mobile\_Number>>t->amount;    if(t->user\_Id==num)  {  cout<<"\t\t\t\tENTER THE AMOUNT YOU WANT TO DEPOSIT :: ";  cin>>amnt;  deposit\_money(amnt);  cout<<"\t\t\t\tNew amount : "<<t->amount;  cout<<"\n\t\t\t\tRecord update \n";  }  if(t==NULL)  {  cout << "\n\n\t\t\t\t<<<<<SORRY Invalid Account no.>>>>>\n";  }  infile.close();  system("cls");  }  if(a==2)  {  fstream infile("bank.txt", ios::binary|ios::in|ios::out);  infile>>t->user\_Id>>t->Name>>t->city>>t->Cnic>>t->mobile\_Number>>t->amount;      if(t->user\_Id==num)  {  cout<<"\t\t\t\tENTER THE AMOUNT YOU WANT TO WITHDRAW :: ";  cin>>amnt;  int balance=t->amount-amnt;  withdrawal(amnt);  cout<<"\t\t\t\tNew amount : "<<balance;    cout<<"\n\t\t\t\trecord updated";  }  if(t==NULL) {  cout << "\n\n\t\t\t\t<<<<<SORRY Invalid Account no.>>>>>\n";  }  infile.close();  system("cls");  }  }  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  // MAIN FUNCTION //  /////////////////////////////////////////////////////////////////  /////////////////////////////////////////////////////////////////  int main()  {  system("color 5E");  head = NULL;  int length;  char password;  string Name, City, Cnic,mobile\_Number;  int amount,new\_amount;  int size=20;  string yes;  int user\_Id,id;    cout<<"\n\n\n\n";  cout<<"\t\t\t\t Loading ";    for(int i=0; i<35; i++)  {  cout<<level;  if(i<10)  Sleep(300);  if(i>=10 && i<20)  Sleep(150);  if(i>=10)  Sleep(25);  }  system("cls");    cout<<" \_\_\_\_ \_ \_\_ \_\_ \_ \_\_\_\_\_ \_ "<<endl;  cout<<" | \_ \ | | | \/ | | | / \_\_\_\_| | | "<<endl;  cout<<" | |\_) | \_\_ \_ \_ \_\_ | | \_\_ | \ / | \_\_ \_ \_ \_\_ \_\_ \_ \_\_ \_ \_\_\_ \_ \_\_ \_\_\_ \_\_\_ \_ \_\_ | |\_ | (\_\_\_ \_ \_ \_\_\_| |\_ \_\_\_ \_ \_\_ \_\_\_ "<<endl;  cout<<" | \_ < / \_` | '\_ \| |/ / | |\/| |/ \_` | '\_ \ / \_` |/ \_` |/ \_ \ '\_ ` \_ \ / \_ \ '\_ \| \_\_| \\_\_\_ \| | | / \_\_| \_\_/ \_ \ '\_ ` \_ \ "<<endl;  cout<<" | |\_) | (\_| | | | | < | | | | (\_| | | | | (\_| | (\_| | \_\_/ | | | | | \_\_/ | | | |\_ \_\_\_\_) | |\_| \\_\_ \ || \_\_/ | | | | | "<<endl;  cout<<" |\_\_\_\_/ \\_\_,\_|\_| |\_|\_|\\_\ |\_| |\_|\\_\_,\_|\_| |\_|\\_\_,\_|\\_\_, |\\_\_\_|\_| |\_| |\_|\\_\_\_|\_| |\_|\\_\_| |\_\_\_\_\_/ \\_\_, |\_\_\_/\\_\_\\_\_\_|\_| |\_| |\_| "<<endl;  cout<<" \_\_/ | \_\_/ | "<<endl;  cout<<" |\_\_\_/ |\_\_\_/ "<<endl;  cout<<" "<<endl;        cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t | DATA STRUCTURES AND ALGORITHMS | "<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;    getch();  system("cls");  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t | GROUP MEMBERS | "<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t | RABIA LATIF 20-SE-024 | "<<endl;  cout<<"\t\t\t\t | M.SADDIQUE 20-SE-010 | "<<endl;  cout<<"\t\t\t\t | AZAN HABIB 20-SE-034 | "<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  getch();  system("cls");  // Menu-driven program  cout<<endl;  cout<<endl;  cout<<endl;  cout<<endl;  while(true){    cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t | WELCOME TO BANK MANAGEMENT SYSTEM | "<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;    cout<<"\t\t\t\t PRESS\n";  cout<<"\t\t\t\t [1]. CREATE ACCOUNT \n";  cout<<"\t\t\t\t [2]. DELETE ACCOUNT \n";  cout<<"\t\t\t\t [3]. ACCOUNT INQUIRY \n";  cout<<"\t\t\t\t [4]. VIEW ALL ACCOUNTS \n";  cout<<"\t\t\t\t [5]. DEPOSIT MONEY \n";  cout<<"\t\t\t\t [6]. WITHDRAW MONEY \n";  cout<<"\t\t\t\t [7]. MODIFY ACCOUNT \n";  cout<<"\t\t\t\t [8]. EXIT \n";  cout<<"\t\t\t\t ENTER YOUR CHOICE ::: ";      int Choice ,i;  char ch;    // Enter Choice  cin >> Choice;  if (Choice == 1) {  ofstream outfile;  outfile.open("bank.txt", ios::out | ios::app |ios::in);  system("cls");      cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*\*CREATE ACCOUNT\*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout <<"\t\t ENTER NAME OF USER:: ";  cin >>Name;  outfile<<Name;  cout <<"\t\t ENTER USER ID:: ";  cin >> user\_Id;  outfile<<user\_Id;  cout <<"\t\t ENTER CITY:: ";  cin >> City;  outfile<<City;    cout <<"\t\t ENTER CNIC:: ";  cin >> Cnic;  outfile<<Cnic;  cout<<"\t\t ENTER AMOUNT:: ";  cin>>amount;  outfile<<amount;  cout<<"\t\t ENTER MOBILE NUMBER:: ";  cin>>mobile\_Number;  outfile<<mobile\_Number;    Insert\_Record(user\_Id, Name, City, Cnic,mobile\_Number,amount);    outfile.close();  }  else if (Choice == 2) {  system("cls");    cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*\* DELETE RECORD \*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;    cout <<"\t\t\t\t ENTER ID OF USER WHOSE RECORD IS TO BE DELETED ";  cin >> user\_Id;  Delete\_Record(user\_Id);  }  else if (Choice == 3) {  system("cls");      cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*\* DELETE RECORD \*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;    cout <<"\t\t\t\t ENTER ID OF USER WHISE RECORD YOU WANT TO SEARCH : ";  cin >> user\_Id;  Search\_Record(user\_Id);  }  else if (Choice == 4) {  system("cls");    cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*\*RECORD DISPLAY\*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  Show\_Record(user\_Id);  }  else if (Choice == 5) {  system("cls");  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*\*DEPOSIT MONEY\*\*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\tENTER THE ID : ";  cin>>user\_Id;  deposit\_withdraw(user\_Id,1);  getch();  }  else if(Choice==6){  system("cls");    cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*WITHDRAW MONEY\*\*\*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\tENTER THE ID : ";  cin>>user\_Id;  deposit\_withdraw(user\_Id,2);  getch();  }  else if(Choice==7){  system("cls");  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*MODIFY ACCOUNT\*\*\*\*\*\*\*\*\*"<<endl;  cout<<"\t\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;  cout<<"\t\t\t\tENTER THE ID : ";  cin>>user\_Id;  modify\_account(user\_Id);  getch();    }  else if(Choice==8){  break;  }  else {  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ "<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t | INVALID CHOICE | "<<endl;  cout<<"\t\t\t\t | TRY AGAIN | "<<endl;  cout<<"\t\t\t\t | | "<<endl;  cout<<"\t\t\t\t ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ "<<endl;  getch();  }    }  } |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*