# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab Quiz # 01

# Task

Write a program of bank management system to manage the account information using inheritance concept.

Create a class “Bank Account” with the customer\_name, account\_number etc. as member variables. Create the derived classes for two types of accounts i.e. current and saving. The derived classes will update the balance and handle the deposit and withdraw cases. Customers should be able to get updated balance after deposit and withdrawal amounts.

**Answer:**

|  |
| --- |
| Solution |
| Task Code:  // stdafx.cpp : source file that includes just the standard includes  // ConsoleApplication1.pch will be the pre-compiled header  // stdafx.obj will contain the pre-compiled type information  #include "stdafx.h"  #include <iostream>  #include <string>  // TODO: reference any additional headers you need in STDAFX.H  // and not in this file  using namespace std;  class BankAccount  {  private:  string customer\_name;  int account\_number;  int amount;  public:  BankAccount()  {  customer\_name = "unknown";  account\_number = 121212;  amount = 200000;  }  BankAccount(string n, int acc, int amnt)  {  customer\_name = n;  account\_number = acc;  amnt = amnt;  }  void setBalance(int n)  {  amount = n;  }  int getBalance()  {  return amount;  }  virtual void withdraw() = 0;  virtual void deposit() = 0;  };  class Current : public BankAccount  {  public:  void withdraw()  {  int n;  cout << "Enter balance you want to withdraw: ";  cin >> n;  int amount = getBalance();  if (n > amount)  {  cout << "Insufficient funds for withdrawal. Current Balance: " << amount << endl;  }  else  {  int newBlnc = amount - n;  setBalance(newBlnc);  cout << "Withdrawal successful. New Balance: " << newBlnc << endl;  }  return;  }  void deposit()  {  int n;  cout << "Enter amount you want to deposit: ";  cin >> n;  int amount = getBalance();  if (n < 0)  {  cout << "Please enter positive integer. Current Balance: " << amount << endl;  }  else  {  int newBlnc = amount + n;  setBalance(newBlnc);  cout << "Deposit successful. New Balance: " << newBlnc << endl;  }  }  };  class Saving : public BankAccount  {  private:  int withdrawLimit = 15000;  public:  void withdraw()  {  int n;  cout << "Enter balance you want to withdraw: ";  cin >> n;  int amount = getBalance();  if (n > amount)  {  cout << "Insufficient funds for withdrawal. Current Balance: " << amount << endl;  }  else if (n > withdrawLimit)  {  cout << "You cannot withdraw more than " << withdrawLimit << ". Current Balance: " << amount << endl;  }  else  {  int newBlnc = amount - n;  setBalance(newBlnc);  cout << "Withdrawal successful. New Balance: " << newBlnc << endl;  }  return;  }  void deposit()  {  int n;  cout << "Enter amount you want to deposit: ";  cin >> n;  int amount = getBalance();  if (n < 0)  {  cout << "Please enter positive integer. Current Balance: " << amount << endl;  }  else  {  int newBlnc = amount + n;  setBalance(newBlnc);  cout << "Deposit successful. New Balance: " << newBlnc << endl;  }  }  };  int main()  {  bool exit = false;  BankAccount\* account;  char choice;  cout << "Enter 1 for Current Account\n";  cout << "Any other key for Saving Account\n";  cin >> choice;  string name;  cout << "Enter Name: ";  cin >> name;  if (choice == '1')  account = new Current();  else  account = new Saving();  while (exit == false)  {  char choice;  cout << "1) Withdraw Amount\n";  cout << "2) Deposit Amount\n";  cout << "3) Check Amount\n";  cout << "4) Exit.\n";  cin >> choice;  switch (choice)  {  case '1':  account->withdraw();  break;  case '2':  account->deposit();  break;  case '3':  cout<<"Your Balance is: " << account->getBalance() << endl;  break;  case '4':  exit = true;  break;  default:  cout << "Select valid option.\n";  break;  }  }  return 0;  }  Task Output Screenshot: |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS.