**Lab Sections**

1. Objectives
2. Introduction
3. Definitions & Important Terms
4. Declaration Syntax
5. Experiments

|  |
| --- |
|  |

Introduction to Classes

**Introduction to Classes**

1. **Objectives**

**After you complete this experiment you will be able to implement a class.**

1. **Introduction**

Classes encapsulate data and the functions that operate on that data. Classes follow the property “I can do everything for myself”.

1. **Definitions & Important Terms**

We will define several terms you need to know to understand classes. They are as follows:

1. A **private member** is a member that can only be accessed while inside the class (within member functions of the class).
2. A **public member** is a member that can be accessed while inside or outside the class.
3. The **state** refers to the private data members.
4. The **behavior** refers to the pubic member functions.
5. Every class member function has access to the **this** pointer.
6. **Mutators** are member functions that change the state of a class.
7. **Accessors** are member functions that do not change the state of a class.
8. **Constructors** initialize the state of the class. Consider the following characteristics:
   * They have the same name as the class;
   * They have no return type;
   * The **default constructor** has no arguments/formal parameters; a class has only one default constructor;
   * The **explicit-value constructor** has arguments/formal parameters; a class can have many explicit-value constructors;
   * The **copy constructor** is used during a call-by-value, in a return statement and in an initialization/declaration statement.
9. **Destructors** de-allocate dynamic memory allocated by the class using the **new** operator.
10. **Helper functions** are private member functions. This means that they can only be used by member functions of the class.
11. The dot operator, “**.**”, is used to access the members of a class.
12. An object is an instance of a class.
13. The scope resolution operator, “**::**”, specifies ownership/membership.
14. **Declaration Syntax**

class Class\_name

{

public:

constructors

destructor

member functions

accessors

mutators

public data

private:

helper functions

data

};

**Example:**

class Bank\_Transaction

{

public:

Bank\_Transaction( ); //initialize the state

double Check\_Balance( ); //return the dollar amount of balance

void Deposit(double); //increase balance by a dollar amount

void Withdrawal(double); //decrease balance by a dollar amount

private:

double balance;

};

More information on classes can be found in your course textbook and on the web.

1. **Experiments**

**Step 1: In this experiment you will investigate the implementation of a class.**

**Enter, save, compile and execute the following program in MSVS. Call the new project “IntroClassesExp1” and the program “IntroClasses1.cpp”. Answer the questions below:**

#include <iostream>

using namespace std;

class Bank\_Acct

{

public:

Bank\_Acct( );

double Check\_Balance( );

void Deposit(double);

void Withdrawal(double);

private:

double balance;

};

Bank\_Acct::Bank\_Acct()

{

balance = 0;

}

double Bank\_Acct::Check\_Balance()

{

return balance;

}

void Bank\_Acct::Deposit(double amount)

{

balance = balance + amount;

}

void Bank\_Acct::Withdrawal(double amount)

{

balance = balance - amount;

}

int main()

{

Bank\_Acct my\_Acct;

cout<<"My Account Balance = "<<my\_Acct.Check\_Balance()<<endl;

my\_Acct.Deposit(2516.83);

cout<<"My Account Balance = "<<my\_Acct.Check\_Balance()<<endl;

my\_Acct.Withdrawal(25.96);

cout<<"My Account Balance = "<<my\_Acct.Check\_Balance()<<endl;

return 0;

}

1. Please list the elements that make up the state of the class “Bank\_Acct” in the program in Step 1?
2. Please list the element(s) that make up the behavior of the class “Bank\_ Acct” in the program in Step 1?
3. What kind of member function is Check\_Balance in the program in Step 1?
4. What kind of member function is Withdrawal in the program in Step 1?
5. What kind of member function is Deposit in the program in Step 1?
6. What kind of member function is Bank\_Acct in the program in Step 1?
7. Can you describe the operation of the dot operation in the program in Step 1?
8. Referring to the first cout statement in the program in Step 1, when was the account balance set to 0? Explain your answer?

**Step 2:** **Enter, save, compile and execute the following program in MSVS. Call the new project “IntroClassesExp2” and the program “IntroClasses2.cpp”. Answer the questions below:**

#include <iostream>

using namespace std;

class Bank\_Transaction

{

public:

Bank\_Transaction( ); //default constructor

Bank\_Transaction(double);

double Check\_Balance( );

void Deposit(double);

void Withdrawal(double);

private:

double balance;

};

Bank\_Transaction::Bank\_Transaction()

{

balance = 0;

}

Bank\_Transaction::Bank\_Transaction(double amount)

{

balance = amount;

}

double Bank\_Transaction::Check\_Balance()

{

return balance;

}

void Bank\_Transaction::Deposit(double amount)

{

balance = balance + amount;

}

void Bank\_Transaction::Withdrawal(double amount)

{

balance = balance - amount;

}

int main()

{

Bank\_Transaction my\_Acct;

Bank\_Transaction your\_Acct(10340.85);

cout<<"Your Account Balance = "<<your\_Acct.Check\_Balance()<<endl;

your\_Acct.Deposit(512.30);

cout<<"Your Account Balance = "<<your\_Acct.Check\_Balance()<<endl;

your\_Acct.Withdrawal(8284.56);

cout<<"Your Account Balance = "<<your\_Acct.Check\_Balance()<<endl;

cout<<"My Account Balance = "<<my\_Acct.Check\_Balance()<<endl;

my\_Acct.Deposit(2516.83);

cout<<"My Account Balance = "<<my\_Acct.Check\_Balance()<<endl;

my\_Acct.Withdrawal(25.96);

cout<<"My Account Balance = "<<my\_Acct.Check\_Balance()<<endl;

return 0;

}

1. Write the statement(ie. the actual line of code) in the program in Step 2 that initializes the balance of the object “your\_Acct”.
2. Give the full name of the function(This is also referred to as the function header) and state the type of the constructor that initialized the object “your\_Acct”.
3. What happens if you add the statement “my\_Acct.Balance = 0;” to the main function of the program in Step 2 after the object declarations? Explain your answer.

**Question 12:** What do we mean when we use the following phases?

a. Inside the class

b. Outside the class