Introduction to Classes

Lab Sections

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

Introduction to Classes

1. Objectives

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

2. Introduction

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

3. 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.

4. 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.

5. 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;
}
```

Question 1: Please list the elements that make up the state of the class "Bank_Acct" in the program in Step 1?

Question 2: Please list the element(s) that make up the behavior of the class "Bank_ Acct" in the program in Step 1?

Question 3: What kind of member function is Check_Balance in the program in Step 1?

Question 4: What kind of member function is Withdrawal in the program in Step 1?

Question 5: What kind of member function is Deposit in the program in Step 1?

Question 6: What kind of member function is Bank_Acct in the program in Step 1?

Question 7: Can you describe the operation of the dot operation in the program in Step 1?

Question 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;
}
```

Question 9: Write the statement(ie. the actual line of code) in the program in Step 2 that initializes the balance of the object “your_Acct”.

Question 10: 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”.

Question 11: 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