|  |
| --- |
| **Computer Engineering Department** |
| Assignment “1” |
| Software Engineering |
|  |
|  |
| **10/19/2015** |

**1. What is a class?**

* The class is the single most important C++ enhancement for implementing these features and tying them together.
* A class is a C++ vehicle for translating an abstraction to a user-defined type. It combines data representation and methods for manipulating that data into one neat package. Let’s look at a class that represents stocks.

**2. How does a class accomplish abstraction, encapsulation, and data hiding?**

* Abstraction is the crucial step of representing information in terms of its interface with the user. That is, you abstract the essential operational features of a problem and express a solution in those terms. In the softball statistics example, the interface describes how the user initializes, updates, and displays the data. From abstraction, it is a short step to the user-defined type, which in C++ is a class design that implements the abstract interface.
* The public member functions act as go-betweens between a program and an object’s private members; they provide the interface between object and program. This insulation of data from direct access by a program is called *data hiding*.

**3. What is the relationship between an object and a class?**

A class is a user-defined type, and an object is an instance of a class. This means an object is a variable of that type or the equivalent of a variable, such as memory allocated by new according to the class specification.

**4. In what way, aside from being functions, are class function members different from class data members?**

Data members go into the private section and member functions go into the public section.

**5. Deﬁne a class to represent a bank account. Data members should include the depositor’s name, the account number (use a string), and the balance.**

Member functions should allow the following:

* Creating an object and initializing it.
* Displaying the depositor’s name, account number, and balance
* Depositing an amount of money given by an argument
* Withdrawing an amount of money given by an argument

Just show the class declaration, not the method implementations.

(Programming Exercise 1 provides you with an opportunity to write the implementation.)

**6. When are class constructors called? When are class destructors called?**

* When a Stock object has three values to be provided from the outside world, you should give the constructor three arguments. (The fourth value, the total\_val member, is calculated from shares and share\_val,so you don’t have to provide it to the constructor.) Possibly, you may want to provide just the company member value and set the other values to zero; you can do this by using default arguments
* When you use a constructor to create an object, the program undertakes the responsibility of tracking that object until it expires. At that time, the program automatically calls a special member function bearing the formidable title *destructor*. The destructor should clean up any debris, so it actually serves a useful purpose. For example, if your constructor uses new to allocate memory, the destructor should use delete to free that memory.

**7. Provide code for a constructor for the bank account class from Chapter Review Question 5.**

**8. What is a default constructor? What is the advantage of having one?**

* A *default constructor* is a constructor that is used to create an object when you don’t provide explicit initialization values.

That is, it’s a constructor used for declarations like this:

Stock fluffy\_the\_cat; // uses the default constructor

* The net result is that the fluffy\_the\_cat object is created with its members uninitialized, just as the following creates x without providing a value for x:

int x;

**9. Modify the Stock class deﬁnition (the version in stock20.h) so that it has member functions that return the values of the individual data members. Note: A member that returns the company name should not provide a weapon for altering the array.**

**That is, it can’t simply return a string reference. It could return a constant reference.**

**10. What are this and \*this?**

You can do still more with the Stock class. So far each class member function has dealt with but a single object: the object that invokes it. Sometimes, however, a method might need to deal with two objects, and doing so may involve a curious C++ pointer called “this”.