Agenda / Learning Objectives:

1. Run the following command and extract lab09.tar in your venus account (note the dot):
   ```
   cp ~ctse/cs211/lab09.tar . ; tar xvf lab09.tar
   ```

2. Check out the textbook resources inside piazza. Also, take a look at key points, tips and pitfalls on page 2 below.

3. Complete the lab exercise for writing constructor below:

   Chapter 7 q7:
   Do Programming Project 6.8, the definition of a Money class, except create a default constructor that sets the monetary amount to 0 dollars and 0 cents, and create a second constructor with input parameters for the amount of the dollars and cents variables. Modify your test code to invoke the constructors.
Key Points + Tips

Constructor Definitions. A constructor is a member function having the same name as the class. The purpose of a class constructor is automatic allocation and initialization of resources involved in the definition of class objects. Constructors are called automatically at definition of class objects. Special declarator syntax for constructors uses the class name followed by a parameter list but there is no return type.

Constructor initialization section. The implementation of the constructor can have an initialization section:

```cpp
A::A(): a(0), b(1) { /* implementation */ }
```

The text calls the :a(0), b(1) the initialization section. In the literature, this sometimes called a member initializer list. The purpose of a member initializer list is to initialize the class data members. Only constructors may have member initializer lists.

Invoking constructors. You cannot call a constructor as if it were a member function, but frequently it is useful to invoke a constructor explicitly. In fact this declaration of class A in object u:

```cpp
A u(3);
```

is short hand for

```cpp
A u = A(3);
```

Here, we have explicitly invoked the class A constructor that can take an int argument. When we need to build a class object for return from a function, we can explicitly invoke a constructor.

```cpp
A f()
{
    int i;
    // compute a value for i
    return A(i);
}
```

Always Include a Default Constructor. A default constructor will automatically be created for you if you do not define one, but it will not do anything. However, if your class definition includes one or more constructors of any kind, no constructor is generated automatically.

Pitfalls

Attempting to invoke a constructor like a member function. We cannot call a constructor for a class as if it were a member of the class.

Constructors with No Arguments. It is important not to use any parentheses when you declare a class variable and want the constructor invoked with no arguments, e.g.

```cpp
MyClass obj;
```

instead of

```cpp
MyClass obj();
```

Otherwise, the compiler sees it as a prototype declaration of a function that has no parameters and a return type of MyClass.

Inconsistent Use of const. If you use const for one parameter of a particular type, then you should use it for every other parameter that has that type and is not changed by the function call.