# CST209

# Object-oriented Programming C++

# (Week 6)

**Dr. Teo Bee Guan**

# Content

- Memberwise Assignment
- **`const`** Objects and **`const`** Member Functions
- Composition: Objects as Members of Classes
- Dynamic members

# Memberwise Assignment

- The assignment operator (=) can be used to assign an object to another object of the same class.

- By default, such assignment is performed by memberwise assignment (also called copy assignment) — each data member of the object on the right of the assignment operator is assigned individually to the same data member in the object on the left of the assignment operator.

**Practice: Example 1**

## In-Class Exercise 1

Create a class named Circle with only one member variable, radius. The class should have only one constructor that accept one argument to initialize radius. Test your class by creating two Circle objects and then assign the first circle object to the second circle object.

# const Objects and const Member Functions

- C++ disallows member function calls for `const` objects unless the member functions themselves are also declared `const`.
- This is true even for get member functions that do not modify the object.
- This is also a key reason that we've declared as `const` all member-functions that do not modify the objects on which they're called.

**Practice: Example 2**

# Composition: Objects as Members of Classes

- It is common for a C++ class to have objects of other C++ classes as data members.
- For example, we might define a Course class to encapsulate a collection of students.
- The new Course could have an array of Student objects as one of its data members.
- This relationship between a Course object and the Student objects that are its data members is called a "has a" relationship. A Course object "has" an array of Student objects.

**Practice: Example 3, 4**

# Composition: Objects as Members of Classes

## In-Class Exercise 2

Create a class named `Employee` with three member variables: `employeeID`, `name`, and `position`. Create another class named `Company` with three member variables: `companyName`, `industry` and `employeeList`. The `employeeList` should hold an array of `Employee` objects (Just set the array size to 3 for simplicity).

Test your class by creating one `Company` object and printing all the three employees' details.

# Dynamic Member

- Dynamic Member is a concept of implementation of user-defined data types which can shrink or grow according to user needs at the runtime of the program.
- The core concept which allows us to create some data containers of variable length is the dynamic memory allocations in C++ programming.
- In general, a pointer to locate the memory blocks, a number count which can represent how much data is stored in the variable, and a maximum size that will indicate that these many elements can be inserted without making any changes to the data container is used in dynamic members.

**Practice: Example 5**

See you next class