# CST209

# Object-oriented Programming C++

# (Week 5)

**Dr. Teo Bee Guan**

# Content

- Overloading Constructor
- Private Member Functions
- Array of Objects
- Instance and Static Members

# Overloading Constructor

- A class can have more than one constructor.
- A class's member functions may be overloaded, including the constructor.
- One constructor might take an integer argument, for example, while another constructor takes a double .

**Practice: Example 1**

## In-Class Exercise 1

- Create a class named Point with two member variables coord_x & coord_y. Try to create three overloaded constructors for this class:
  i. The first constructor does not take any parameter. The private members coord_x and coord_y should be set to zero as default value.
  ii. The second constructor should take only one parameter x to initialize the coord_x and set the coord_y to zero as default value.
  iii. The third constructor should take two parameter x and y to initialize the coord_x and coord_y.

  Test your class by creating three Point objects.

# Private Member Functions

- Sometimes a class will contain one or more member functions that are necessary for internal processing, but should not be called by code outside the class.

- For example, a class might have a member function that performs a calculation only when a value is stored in a particular member variable and should not be performed at any other time.

- A private member function can only be called from a function that is a member of the same class.

# Private Member Functions

- When a member function is declared private , it may only be called internally.

- That function should not be directly accessible by code outside the class

**Practice: Example 2**

## In-Class Exercise 2

- Extend your code from Example 2, add two more private member variable, ***`idNumber` (int) and `passID` (string).*** Create another private function, `generatePassID`, by joining the last name with the `idNumber`.

Test your class by creating three Person objects.

```
John
Smith
John Smith
24577
Smith24577

Helen
Keller
Helen Keller
98833
Keller98833

Jack
Babara
Jack Babara
63568
Babara63568
```

Sample Output

# Arrays of objects

- As with any other data type in C++, we can define arrays of class objects.
- Here is an example of such a definition:

```
const int ARRAY_SIZE = 3;
Inventory inventoryItem[ARRAY_SIZE];
```

**Practice: Example 3**

## In-Class Exercise 3

- Extend your code from the Exercise 2, create an array of Person objects with constant size 3.

  Test your class by printing each Person object values in a for loop.

```
John
Smith
John Smith
24577
Smith24577

Helen
Keller
Helen Keller
98833
Keller98833

Jack
Babara
Jack Babara
63568
Babara63568
```

Sample Output

# Instance and Static Members

- Each instance of a class has its own copies of the class's instance variables.
- If a member variable is declared static , however, all instances of that class have access to that variable.
- If a member function is declared static , it may be called without any instances of the class being defined.

**Practice: Example 4**

## In-Class Exercise 4

- Extend your code from the Exercise 3, add a static member variable named as `personCount` and add another static method named as `getPersonCount`.

Test your static members by creating three Person objects.

See you next class