FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

# Encapsulation

## CSPROG2
## Computer Programming 2 for CS

# Specific Objective

- Learn about the definition of encapsulation
- Discuss the different derive class that can restrict visibility
- Differentiate the function of the four Important Operation in Encapsulation
- Learn the proper way in naming encapsulation

# Encapsulation

**It allows the programmer to group data and the subroutines that operate on them together in one place, and to hide irrelevant details from the user.**

# Encapsulation

**Derived class can restrict visibility**

- **Private**
  - **Protected and public members of base class are private in derived class.**

# Encapsulation

## Derived class can restrict visibility

- **Protected**
  - **Protected and public members of base class are protected in derived class.**

# Encapsulation

**Derived class can restrict visibility**

- **Protected and public members of base class are protected and public in derived class.**

# Encapsulation

**Derived class can restrict visibility**

- **Public**
- **Private members of base class aren't visible in derived class.**

# Encapsulation

## Four Important Operation in Encapsulation:

- Choosing a Constructor
- References and Values
- Execution Order
- Garbage Collection

# Encapsulation

## Choosing a Constructor

- Object-Oriented Languages allow classes to have zero, one or more different constructors.
- Two ways to distinguish between constructors
  - Different Names
  - Different Number and Types of Arguments

# Encapsulation

## References and Values

- Reference
  - Every object is created explicitly so it is easy to make sure the correct constructor is called.
  - It requires allocation from heap and extra indirections on every access of the object.
- Value
  - More efficient but harder to control initialization

# Encapsulation

## Execution Order

- If class B is derived from class A, A constructor is called before B constructor
  - To get arguments to the A constructor, you must use an intializer list
  - The part after the colon is a call to bar's constructor

# Encapsulation

## Garbage Collection

- When an object is destroyed, the destructor is called for the derived class first, then the destructors of the base classes are called.
- Many languages provide automatic garbage collection

# Encapsulation

**Encapsulation in C++**

- Can define header and code files,
- Or, classes can be used for encapsulation
  - The class header file has only the prototypes of the member functions
  - The member definitions are defined in a separate file

# Encapsulation

## Naming Encapsulations

- C++ **namespaces**
  - Can place each library in its own namespace and qualify names used outside with the namespace

```
namespace MyStack {
  ...   // stack declarations      }
```

# Encapsulation

## Naming Encapsulations

Can be referenced in three ways:

```
MyStack::topPtr
using MyStack::topPtr;        p = topPtr;
using namespace MyStack;      p = topPtr;
```

– C# also includes namespaces