



Module 10: Review

Dr. Tran Minh Triet

Acknowledgement

❖ Slides

- Course CS202: Programming Systems
Instructor: MSc. Karla Fant,
Portland State University
- Course CS202: Programming Systems
Instructor: Dr. Dinh Ba Tien,
University of Science, VNU-HCMC
- Course DEV275: Essentials of Visual Modeling with
UML 2.0
IBM Software Group

Outline

- ❖ Main concepts of OOP
- ❖ Principles of OOP
- ❖ Class declaration, visibility of member attributes/functions
- ❖ Types of member functions
- ❖ Operators
- ❖ Constructor, destructor, copy assignment
- ❖ Inheritance, types of inheritance
- ❖ Polymorphism
- ❖ Relationships between classes

Main concepts of OOP

- ❖ Class
- ❖ Object
- ❖ State → Attribute
- ❖ Behavior → (Member) function

Principles of OOP

- ❖ Abstraction
- ❖ Encapsulation
- ❖ Modularity
- ❖ Hierarchy

cuu duong than cong . com

Class declaration, visibility of member attributes/functions

❖ Class declaration and definition (implementation)

❖ Visibility:

- Public
- Protected
- Private

Types of member functions

- ❖ Constructor/Initialization
- ❖ Observer
- ❖ Mutator
- ❖ Iterator

cuu duong than cong . com

cuu duong than cong . com

Operators

❖ Operator overloading

cuu duong than cong . com

cuu duong than cong . com

Constructor, destructor, copy assignment

- ❖ When a constructor is invoked?
- ❖ When a destructor is invoked?
- ❖ Sequence of invoked constructors when creating an object of a derived class
- ❖ Sequence of invoked destructors when destroying an object of a derived class
- ❖ Copy constructor
- ❖ Copy assignment

Inheritance, types of inheritance

- ❖ Inheritance, the semantic of generalization relationship
- ❖ Type of inheritance:
 - Public
 - Protected
 - Private

Inheritance, types of inheritance

- ❖ Initializing a pointer to an object of a base class
- ❖ Typecasting a pointer to an object of a base class
- ❖ Actual type and formal type of a pointer to an object

Polymorphism

❖ Which function will be invoked?

- Formal type of a pointer
- Actual type of a pointer

❖ Applications

Relationships between classes

❖ Generalization

❖ Association

- Aggregation
- Composite

❖ Dependency