POSSIBLE TOPICS for FINAL

21st LECTURE

엄현상(Eom, Hyeonsang)
School of Computer Science and Engineering
Seoul National University

Outline

- **C++ Code Examples**
- **Possible Topics**

Code Examples

- Visitor

```
#include <iostream>

using namespace std;

class Item;
class ItemA;
class ItemB;

class ItemVisitor;
class Pricing1Visitor;
class Pricing2Visitor;
```

"Design Patterns: Elements of Reusable Object-Oriented Software,"Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class Item {
public:
        virtual ~Item() {};

        void SetPrice(double p) { _price = p; };
        double GetPrice() { return _price; };

        virtual double NetPrice() { return _price; };
        virtual double DiscountPrice() { return 0.6*_price; };

        virtual void Accept(ItemVisitor&) = 0;
protected:
        Item() {};
private:
        double _price;
};
```

"Design Patterns: Elements of Reusable Object-Oriented Software,"Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class ItemA : public Item {
public:
        ItemA() {};
        virtual void Accept(ItemVisitor& v) { v.VisitItemA(this); }
};

class ItemB : public Item {
public:
        ItemB() {};
        virtual void Accept(ItemVisitor& v) { v.VisitItemB(this); }
};
```

"Design Patterns: Elements of Reusable Object-Oriented Software,"Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class ItemVisitor {
public:
        virtual void VisitItemA(Item*) {};
        virtual void VisitItemB(Item*) {};
protected:
        ItemVisitor() {};
};
```

"Design Patterns: Elements of Reusable Object-Oriented Software,"Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class Pricing1Visitor : public ItemVisitor {
public:
   Pricing1Visitor() { _total = 0.0; };

            double GetTotalPrice() { return _total; };

            virtual void VisitItemA(ItemA* pa) { _total += pa->NetPrice(); };
            virtual void VisitItemB(ItemB* pb) { _total += pb-
>DiscountPrice(); };

private:
            double _total;
};
```

"Design Patterns: Elements of Reusable Object-Oriented Software,"Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
class Pricing2Visitor : public ItemVisitor {
public:
    Pricing2Visitor() { _total = 0.0; };

            double GetTotalPrice() { return _total; };

            virtual void VisitItemA(ItemA* pa) { _total += pa-
>DiscountPrice(); };
            virtual void VisitItemB(ItemB* pb) { _total += pb->NetPrice(); };

private:
            double _total;
};
```

"Design Patterns: Elements of Reusable Object-Oriented Software,"Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison Wesley, 1995

Code Examples Cont'd

- Visitor Cont'd

```
int main () {
        ItemA a;
        a.SetPrice(10);
        ItemB b;
        b.SetPrice(20);

        Pricing1Visitor p1;
        a.Accept(&p1); // p1.VisitItemA(&a);
        b.Accept(&p1); // p1.VisitItemB(&b);
        cout << "Pricing: p1.GetTotalPrice() = for a & b"
<< p1.GetTotalPrice() << endl;

        Pricing2Visitor p2;
         a.Accept(&p2); // p2.VisitItemA(&a);
         b.Accept(&p2); // p2.VisitItemB(&b);
        cout << "Pricing: p2.GetTotalPrice() = for b & b"
<< p2.GetTotalPrice() << endl;
}
```

Possible Topics about C++

- Namespaces
- Imperative Language (vs. Declarative Lang.)
- Object-Oriented Programming Design Principles
- Separating Interface from Implementation
- Storage Classes and Scope Rules
- Function Overloading and Templates
- C++ Pointers
- friend Functions and Classes
- Inheritance
- Exception Handling

Possible Topics about Java

- Differences between Java and C/C++
- Java Program Execution
- Member Access Control
- Static Modifier
- Locating Classes
- Classes, Objects, and Packages
- Methods & Operators
  - Method Overloading
  - Initialization Order
- Java I/O Systems

Possible Topics about Java (Cont'd)

- Composition vs Inheritance
- Inheritance
  - Initialization
- Final
- Polymporphism
  - Upcasting
  - Downcasting
- Interface
- Exception Handling
- String, Formatter, and Scanner
- Container and Java Tools