Slides #14
Object Oriented Design

Sections 7.9, 7.10, 7.15

CMPT 125/128
© Dr. B. Fraser

Topics

1) Can we pass objects to functions?

2) How can we design object-oriented programs?

Objects and functions

Passing objects to functions

- Objects can be arguments to functions using either:
  - Pass by value
    -
      the object is created and passed to the function.
    - Changes to the object in the function...

  - Pass by reference
    -
      object is passed to the function.
    - Changes to the object in the function...

Review

- What does the following code output?

```
// Includes and prototypes go here..
int main() {
    Circle myCircle(5);
    changeA(myCircle);
    display(myCircle);
    changeB(myCircle);
    display(myCircle);
    ...
}
void display(Circle c) {
    cout << "r: " << c.getRadius() << endl;
}
void changeA(Circle &c) {
    c.setRadius(10);
}
void changeB(Circle c) {
    c.setRadius(42);
}
```

```
class Circle {
private:
    double radius;
public:
    Circle(double r);
    void setRadius(double r);
    double getRadius();
};
```

Output:

Object oriented design/analysis
(OOD / OOA)

Object-oriented analysis

- Object-oriented analysis:

    - Crucial to do before just writing code.
      Try building a house without a plan!

- OOA Steps:
    - Identify the
    - Define   for each class.
    - Define   for each class.
    - Define   between classes.

Step 1: Identifying classes

- Classes are attributes and behaviours put together.
    -

- Restaurant Example:
    - Menu item class: represents an item on the menu.
    - Order class: able to store a list of menu items
        - *"I'll have a big mac, large fry, and caviar."*

- Possible objects:
    - Real world entities   (menu items).
    -                        (customer, employee, boss)
    - IO devices   (keyboard, mouse, screen, ...)

Object Identification

- Identify nouns in the requirements document.
  - Noun: person, place, or thing.

-

  to get an idea for possible classes.

  When customers call to report a product's defect, the user must record: product serial number, what the defect is, and its severity.

Classes are Singular

- Class names are usually...
  - Modelling people   =   class
  - Modelling cars   =   class

- Create multiple objects of the same class for plural:
  - "We rent many cars" --> Multiple instances of Car .

- One object could manage other objects:
  - CarManager for tracking all the Car objects.
  - AccountList for tracking all the Account objects.
  - May want only...

Step 2: Defining class attributes

-

  the program needs for each class.

- Restaurant Example:
  - Menu Item:
    - name, cost, category (desert, drink, entree)
  - Order:
    - order number, time placed, list of items, total.

Key Question:

- Examples:
  - Product barcode:
    - an int, or a custom class?
  - Facebook relationship status:
    - string, or custom class?
- Answer depends on what your requirements are.

-
  - If a class only stores a single value (Ex: a long), it may be a better attribute than a class.

Step 3: Define class behaviours

- Identify the behaviours that the classes...

- Restaurant Example:
  - Menu Item Class
    - Get name, get cost, change price
  - Order Class
    - Add item, calculate total, print bill.

-
  - getX(), setX(), calculateX(), joinX(), display()
  - Start with biggest features, then move to smaller ones.

Assigning behaviours to classes

-
  - Ex: enrolling a student in a course:
    - Member function of Student class?
    - Member function of Course class?
    - Member function of EnrollmentManager class?

- Consider all possibilities then pick the best
  - Revise as needed during design and coding.
  - Ex: Probably put it in the class that stores which students are enrolled in which courses.

Review

- Use OOD to design a cell-phone.
  - Create 3 member variables (attributes).
    - Are any complex enough to be classes?
  - Create 3 member functions (methods).

Step 4: Class relationships

-
  - Where a class "uses" another class.
  - Ex: Any of our programs using cin and cout.

-
  - Where a class "has-a" object of another class in it.
  - Ex: Car has-an Engine.

-
  - Where a class "is-a" sub-category of another class.
  - Ex: Eagle is-a Bird.

Dependency

- Where...
    - Class A calls member functions of class B.
    - Examples
        - Using cout to output.

Class A

```
someMethod(){
    B.aMethod();
}
```

Calls or Uses

Class B

```
aMethod() {

}
```
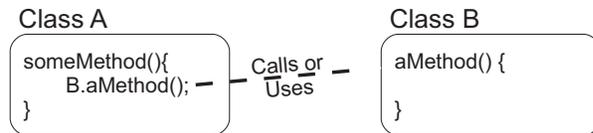
Composition

- Composition:

    - Ex: class A has member variables which are of type class B (and C, ...)

Object A

"has-a"

Objects B, C, ...

Object A

B   C   . . .
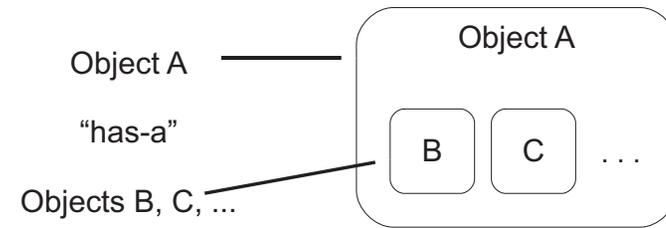
# Composition example

- A class's member variable can be an...

    - Sometimes useful to nest one object inside another.

```
class CycleWheel {
private:
    int size;
    ...
public:
    void setSize(int sz)
    { size = sz; }
    ...
};
```

```
class UniCycle {
private:
    CycleWheel wheel;
    ...
public:
    void setWheelSize(int s)
    { wheel.setSize(s); }
    ...
};
```

```
int main() {
    UniCycle c1;
    c1.setWheelSize(10);
    ...
}
```

UML Composition

- Composition shown with...
  on the containing class.

- Note strings are not shown
  as objects in UML.
    - They are so common that
      they are just treated like
      other data (int, double, ...)

Student
- homeAddress : Address - schoolAddress : Address
+ toString() : string

Address
- streetAddress : string - city : string
+ displayAddress() : void ...
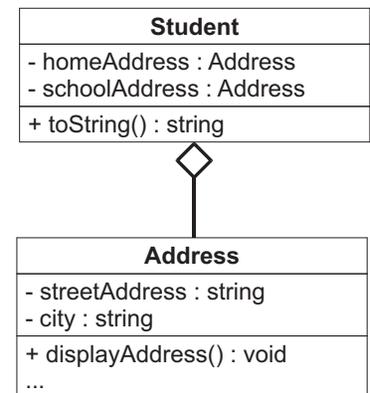
Inheritance

- Inheritance
  - When a class is a subclass of another it creates an..

- Examples
  - Dolphin is a Mammal
  - DigitalCamera is a Camera
  - Apple is a Fruit
- Discussed in later courses.

Summary

- Can pass objects by value (copy) or by reference.
- Object Oriented Analysis/Design:
  - Find the classes (nouns)
  - Find the attributes
  - Find the behaviours (verbs)
  - Find the relationships
    - use,
    - composition,
    - inheritance.