

OO Review

- Objects
- Messages
- Classes
- Inheritance
- Method Resolution
- Polymorphism

1

Objects

- An object is something that can be conceived of.
 - A physical object (book, table, etc.).
 - Or nonphysical (a birthday, an alphabet).
- Objects have state.
 - Described by the values of their attributes at any point in time.
 - The state can change.
- Objects have operations.
 - The operations are the only means by which the state of an object may be changed.
 - Also called methods or member functions.

2

Messages

- An operation is invoked when the object receives a message.
- Objects collaborate by sending messages.
- The message contains the name of the operation and possibly some parameters (this is very similar to a function call, and in some languages is implemented as a function call).
- The only way to change the state of an object is to send it a message.

3

Classes (1 of 2)

- Objects that are similar in some way may be thought of as being in the same class.
- A class is a template for objects.
- Sometimes an object is referred to as an object instance of a class.
- One class may have multiple object instances.
- Creating objects is one of the operations that classes have (to create a new object, you send a message to a class).

4

Classes (2 of 2)

- In code, since all of the objects of the same class have the same operations, there is only one copy of the code for these operations (it is in the class).
- Since each object may have its own state, the attributes are stored in each object.
 - The types of the attributes for all objects of the same class are the same.
 - This type information is in the class, too.

5

Inheritance (1 of 3)

- If two or more classes have similar properties (attribute types or operations), they may be considered to be of the same class.
 - That is, a class of classes.
- There may be multiple subclasses for each superclass.
- This results in a hierarchy of classes.
- There is usually only one root for this hierarchy.

6

Inheritance (2 of 3)

- In some languages (e.g., C++), each class may have multiple superclasses (this is called multiple inheritance).
- In other classes (e.g., Smalltalk, Java), each class may be the subclass of only one superclass (single inheritance).

7

Inheritance (3 of 3)

- The code for operations that are common to all of the subclasses of a superclass is stored in the superclass.
- The type information for the attributes of all of the subclasses of a superclass are stored in the superclass.
- A subclass may have attributes and operations in addition to those it inherits from its superclass.

8

Method Resolution (1 of 3)

- When you send a message to an object, the code that executes may or may not be stored in the class of which that object is an instance.
- The code may be stored in the superclass.
- Or in its superclass.
- Or anywhere up the hierarchy.

9

Method Resolution (2 of 3)

- If the code for the operation is stored in one of the object's superclasses, then the only state information that the operation may manipulate is that which is known about at the superclass level.
- Superclass operations have no knowledge of subclass attributes.

10

Method Resolution (3 of 3)

- Finding the code for an operation is called method resolution (or sometimes called operation resolution).
- If this can be done at compile time, it will make the code run more efficiently. This is called static binding.
- If it is done when the operation is invoked (at run time), it results in greater flexibility. This is called dynamic binding.

11

Polymorphism (1 of 4)

- When multiple objects are able to receive the same message and have an operation that will carry out the desired action, this is called polymorphism (Greek for "many shapes").
- In Smalltalk, all that is necessary is for multiple classes to have methods with the same signature defined.
- In Java, these classes must all be descendants of a common type that defines the common method signature.

12

Polymorphism (2 of 4)

- In Smalltalk, a variable may contain a reference to any object.
- In Java, all variables must be declared using a type definition.
 - If the type is primitive, the value is stored in the variable.
 - If the type is a class, the variable will contain a reference to an object instance.

13

Polymorphism (3 of 4)

- An object reference variable may contain a reference to an object instance of the class of the variable, or any of its descendant classes (but not the other way around).
- However, the only methods that may be invoked are those that are defined by the class of the variable (or its ancestors).

14

Polymorphism (4 of 4)

- Polymorphism depends on dynamic binding of methods.
- At compile time, the compiler checks to see that a message being sent to an object referenced by a variable is legal according to the data type of the variable.
- At run time, the code that executes is determined by the object instance that receives the message.

15