

OBJECTS AND CLASSES

OOP

- OOP (Object Oriented Programming) is the dominant programming paradigm.
- Your program is made of objects, with certain properties and operations that the objects can perform.

THE VOCABULARY OF OOP

- Class
- Object
- Instance
- Instance fields
- Methods
- State
- Extends
- Implements
- ...

OBJECTS

- The object's behavior.
- The object's state.
- The object's identity.

USING EXISTING CLASSES

- `java.lang.String`
- `java.lang.System`
- `java.lang.Exception`
- `java.util.ArrayList`
- `java.util.HashMap`
- `java.lang.Object`
- `java.lang.Thread`
- `java.util.Date`
- ...

BUILDING YOUR OWN CLASSES

```
Class NameOfClass{  
    constructor1  
    constructor2  
  
    ...  
    method1  
    method2  
  
    ...  
    field1  
    field2  
  
    ...  
}
```

BUILDING EMPLOYEE CLASS

- Name.
- Salary.
- HiredDate.
- Constructors.
- Get/set methods.
- RaiseSalary.

CONSTRUCTORS

- A constructor has the same name as the class.
- A class can have more than one constructor.
- A constructor may take zero, one, or more parameters.
- A constructor has no return value.
- A constructor is always called with the new operator.

ACCESS CONTROL MODIFIERS

- Visible to the package, the default. No modifiers are needed.
- Visible to the class only (private).
- Visible to the world (public).
- Visible to the package and all subclasses (protected).

NON ACCESS CONTROL MODIFIERS

- The static modifier for creating class methods and variables.
- The final modifier for finalizing the implementations of classes, methods, and variables.
- The abstract modifier for creating abstract classes and methods.
- The synchronized and volatile modifiers, which are used for threads.

METHOD ACCESS TO PRIVATE DATA

- You know that a method can access the private data of the object on which it is invoked. What many people find surprising is that a method can access the private data of all objects of its class.

PRIVATE METHODS

- These methods can be called only from other methods of the same class.
- The reason is simple: to implement certain methods, you may wish to break up the code into many separate methods. Some of these internal methods may not be particularly useful to the public.

FINAL INSTANCE FIELDS

- You can define an instance field as final. Such a field must be initialized when the object is constructed. That is, it must be guaranteed that the field value is set after the end of every constructor. Afterwards, the field may not be modified again.

STATIC FIELDS

- If you define a field as static, then there is only one such field per class. In contrast, each object has its own copy of all instance fields.

STATIC METHODS

- Static methods are methods that do not operate on objects.
- When a method doesn't need to access the object state because all needed parameters are supplied as explicit parameters.
- When a method only needs to access static fields of the class.

METHOD PARAMETERS

- Call by value.
- Call by reference.

PACKAGES

- Java allows you to group classes in a collection called a package. Packages are convenient for organizing your work and for separating your work from code libraries provided by others.
- A class can use all classes from its own package and all public classes from other packages.