Java Reflection Mechanism

The reflection API represents

- The classes
- Interfaces
- Objects

in the current Java Virtual Machine.

Use the reflection API for Writing development tools such as

- Debuggers
- Class browsers
- GUI builders.

Package to be used

java.lang.reflect.*

What can be done using the Reflection API?

- Determine the class of an object.
- Get information about a class's modifiers, fields, methods, constructors, and superclasses.
- Find out what constants and method declarations belong to an interface.
- Create an instance of a class whose name is not known until runtime.
- Get and set the value of an object's field, even if the field name is unknown to the program until runtime.
- Invoke a method on an object, even if the method is not known until runtime.
- Create a new array, whose size and component type are not known until runtime, and then modify the array's components.

Steps for applying reflection

For each class, the Java Runtime Environment (JRE) maintains an immutable Class (java.lang.Class) object.

Get access to this Class object using :

- Class c = mystery.getClass(); (when instance of the class is available)
- Class c = java.awt.Button.class; (when the name of the class is known at compile-time)
- Class c = Class.forName(strg); (If the class name is unknown at compile time, but available at runtime)

Steps for applying Reflection continued

Use the methods in java.lang.

Class to reflect the internals of the class.