1. Demonstration of a custom annotation.

```
Import java.lang.annotation.* ;
import java.lang.reflect.*;


@Retention(RetentionPolicy.RUNTIME)
@interface MyAnno {
  String str();
  int val();
}

public class Meta {

  // Annotate a method.
  @MyAnno(str = "Annotation Example", val = 100)
  public static void myMeth() {
    Meta ob = new Meta();

    // Obtain the annotation for this method
    // and display the values of the members.
    try {
      // First, get a Class object that represents
      // this class.
      Class c = ob.getClass();

      // Now, get a Method object that represents
      // this method.
      Method m = c.getMethod("myMeth");

      // Next, get the annotation for this class.
      MyAnno anno = m.getAnnotation(MyAnno.class);

      // Finally, display the values.
      System.out.println(anno.str() + " " + anno.val());
    } catch (NoSuchMethodException exc) {
       System.out.println("Method Not Found.");
    }
  }

  public static void main(String args[]) {
    myMeth();
  }
}
```

The result:

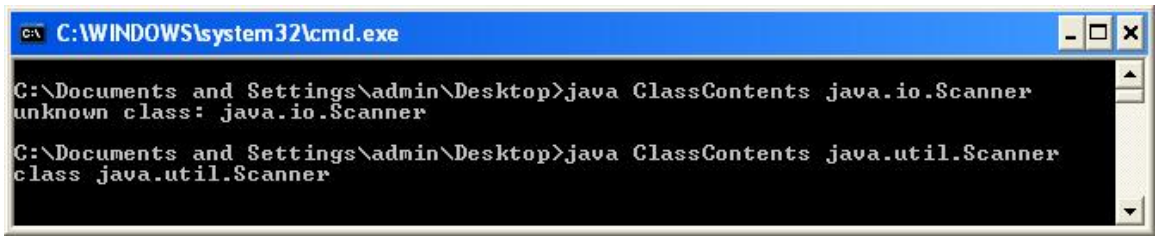
2. Demonstrate a simple use of Class class.

```
import java.lang.reflect.*;

  public class ClassContents {
     public static void main(String[] args) {
         try {
             Class c = Class.forName(args[0]);
             System.out.println(c);
             printMembers(c.getFields());
             printMembers(c.getConstructors());
             printMembers(c.getMethods());
         } catch (ClassNotFoundException e) {
             System.out.println("unknown class: " + args[0]);
         }
     }

     private static void printMembers(Member[] mems) {
         for (Member m : mems) {
             if (m.getDeclaringClass() == Object.class)
                 continue;
             String decl = m.toString();
             System.out.print("    ");
         }
     }
}
```

The result:



Do It Yourself

6.1. Do workshop of the module 8, 9.

6.2. In practical, code of a program is implemented by some members of a project group. It costs much time when manager of the project wants to track work portion of each member. In order to solve this problem, he should build an annotation for methods and

force every member to insert this annotation before his method instead of comment lines. You're in role of the manager and let you develop the own annotation named .

Data members of the annotation: method name, description, author, created date.

6.3. Write a program that takes the name of a Java class as a command-line argument and uses the Class class to print out all the superclasses of that class. For example, if invoked with the argument "java.awt.Applet", the program prints the following: java.lang.Object java.awt.Component java.awt.Container java.awt.Panel.

6.3. Create a class that includes fields: Name, Id, Price and methods: default constructor, two-parameter constructor, get/set methods, display() and toString() methods

Write a reflection program to display fields of class Car (number of fields, field name, type of field). Displaying information of methods (except constructors) (method name, parameters of method).

Executing method setPrice and and print by calling invoke() of class Method. Display fields value after changing.

Set field value of class Car by using set() of class Field.

Building a menu a run corresponding option.

Menu

---

1. Display Mehods
2. Display Fields
3. Set Fields
4. Display All
5. Exit

Your choice: _

References

+ Java tutorials

+ Javadoc

+ Java2s.com

+ Javapassion.com

+ Java almanac
http://www.exampledepot.com