**Lab 3** Intro OOP: Class and Object

**977-210** Object-Oriented Programming

**Part 1: Structural programming and OOP**

Code from slide. Learn the different of programming styles.

1. Structural programming

/\* filename: StructuralClass.java \*/

public class StructuralClass {  
  
 static int *globalVariable* = 10;  
 int nonStaticVariable = 5;  
  
 public static void initMethod(){  
 System.*out*.println("method: initialMethod");  
 System.*out*.println("global\_var = " + *globalVariable*);  
  
 }  
  
 public static void increaseMethod(){  
 int localVariable = 10;  
 *globalVariable* += 10;  
  
 System.*out*.println("Method: increaseMethod");  
 System.*out*.println("global\_var = " + *globalVariable*);  
 System.*out*.println("local\_var = " + localVariable);  
 System.*out*.println();  
  
 }  
 public static void main(String[] args) {  
 System.*out*.println("Structural Programming Style");  
 System.*out*.println("globalVariable = "+ *globalVariable*);  
 *// System.out.println("localVariable = "+ localVariable);  
 // System.out.println("nonStaticVariable = "+ nonStaticVariable);  
  
 initMethod*();  
 System.*out*.println("---------------");  
 *increaseMethod*();  
 System.*out*.println("---------------");  
 *increaseMethod*();  
 System.*out*.println("---------------");  
  
 }  
}

---------------------------------------------------------------------------

1. OOP

/\*filename: ObjectClass.java \*/

public class ObjectClass {  
  
 static int *globalVariable* = 10;  
 int nonStaticVariable = 5;  
  
 public void initMethod(){  
 System.*out*.println("method: initialMethod");  
 System.*out*.println("global\_var = " + *globalVariable*);  
  
 }  
  
 public void increaseMethod(){  
 int localVariable = 10;  
 *globalVariable* += 10;  
  
 System.*out*.println("Method: increaseMethod");  
 System.*out*.println("global\_var = " + *globalVariable*);  
 System.*out*.println("local\_var = " + localVariable);  
 System.*out*.println();  
  
 }  
 public static void main(String[] args) {  
  
 System.*out*.println("Object-Oriented Programming Style");  
  
 ObjectClass obj = new ObjectClass();  
 System.*out*.println("globalVariable = "+ *globalVariable*);  
 System.*out*.println("obj.globalVariable = "+ obj.*globalVariable*);  
 *// System.out.println("localVariable = "+ localVariable);  
 // System.out.println("nonStaticVariable = "+ nonStaticVariable);* System.*out*.println("obj.nonStaticVariable = "+ obj.nonStaticVariable);  
  
 obj.initMethod();  
 System.*out*.println("---------------");  
 obj.increaseMethod();  
 System.*out*.println("---------------");  
 obj.increaseMethod();  
 System.*out*.println("---------------");  
  
 ObjectClass obj2 = new ObjectClass();  
 System.*out*.println("obj2.globalVariable = "+ obj2.*globalVariable*);  
 System.*out*.println("obj2.nonStaticVariable = "+ obj2.nonStaticVariable);  
  
 obj2.initMethod();  
 System.*out*.println("---------------");  
 obj2.increaseMethod();  
 System.*out*.println("---------------");  
 }  
}  
-----------------------------------------------------------

3. OOP: Separate files (template class + client program)

/\* filename: ObjectClass.java \*/

public class ObjectClass {  
  
 static int *globalVariable* = 10;  
 int nonStaticVariable = 5;  
  
 public void initMethod(){  
 System.*out*.println("method: initialMethod");  
 System.*out*.println("global\_var = " + *globalVariable*);  
  
 }  
  
 public void increaseMethod(){  
 int localVariable = 10;  
 *globalVariable* += 10;  
  
 System.*out*.println("Method: increaseMethod");  
 System.*out*.println("global\_var = " + *globalVariable*);  
 System.*out*.println("local\_var = " + localVariable);  
 System.*out*.println();  
  
 }

}

------------------------------------

/\* filename: MyProgram.java \*/

public class MyProgram {  
 public static void main(String[] args) {  
  
 System.*out*.println("Object-Oriented Programming Style");  
  
 ObjectClass obj = new ObjectClass();  
 *// System.out.println("globalVariable = "+ globalVariable);* System.*out*.println("globalVariable = "+ obj.*globalVariable*);  
  
 *// System.out.println("localVariable = "+ localVariable);  
 // System.out.println("nonStaticVariable = "+ nonStaticVariable);* System.*out*.println("nonStaticVariable = "+ obj.nonStaticVariable);  
  
 obj.initMethod();  
 System.*out*.println("---------------");  
 obj.increaseMethod();  
 System.*out*.println("---------------");  
 obj.increaseMethod();  
 System.*out*.println("---------------");  
  
 }  
}

**Part 2: Understand the static keyword**

1. Create class Cube which consist of attributes for calculating the surface area and volume.

* Create method calculateArea() and calculateVolume() for calculating surface area and volume in the classes and print on the screen.
* In main method, create a Cube object with the edge = 5. Call methods to print the surface area and volume.

1. Create class Cylinder which consist of attributes for calculating the surface area and volume.

* Create a static method to calculateArea() and calculateVolume() to calculate surface area and volume in the classes and print on the screen.
* Add main method to get Cylinder variables from user.
* Call methods to print the surface area and volume if necessary.

**Part 3 Review class and object**

Write Employee class with following attributes and methods

|  |
| --- |
| **Employee** |
| + id : int  + name : String  + position : String  + salary : double |
|
| + getEmployeeDetail() : print all attributes  + calculateTax(double taxRate) : calculate tax and print salary, tax, and net salary |

Write a java program to create an Employee object. Get object variables from the keyboard and call all methods.

Output:

\*\*\*\* Payroll System \*\*\*\*

Enter employee detail

id: **1234**

name: **Micheal K**.

position: **Programmer**

salary: **50000**

Enter tax rate(%): **10**

==============================

ID Name Position

1234 Micheal K. Programmer

==============================

Salary: 50000.0

Tax 10%: 5000.0

Net Salary: 45000.0

==============================