Java Programming AP Edition
U3C9 Objects and Classes

OVERVIEW OF CLASS AND OBJECTS FROM PROGRAM STRUCTURE POINTS OF VIEW

ERIC Y. CHOU, PH.D. IEEE SENIOR MEMBER

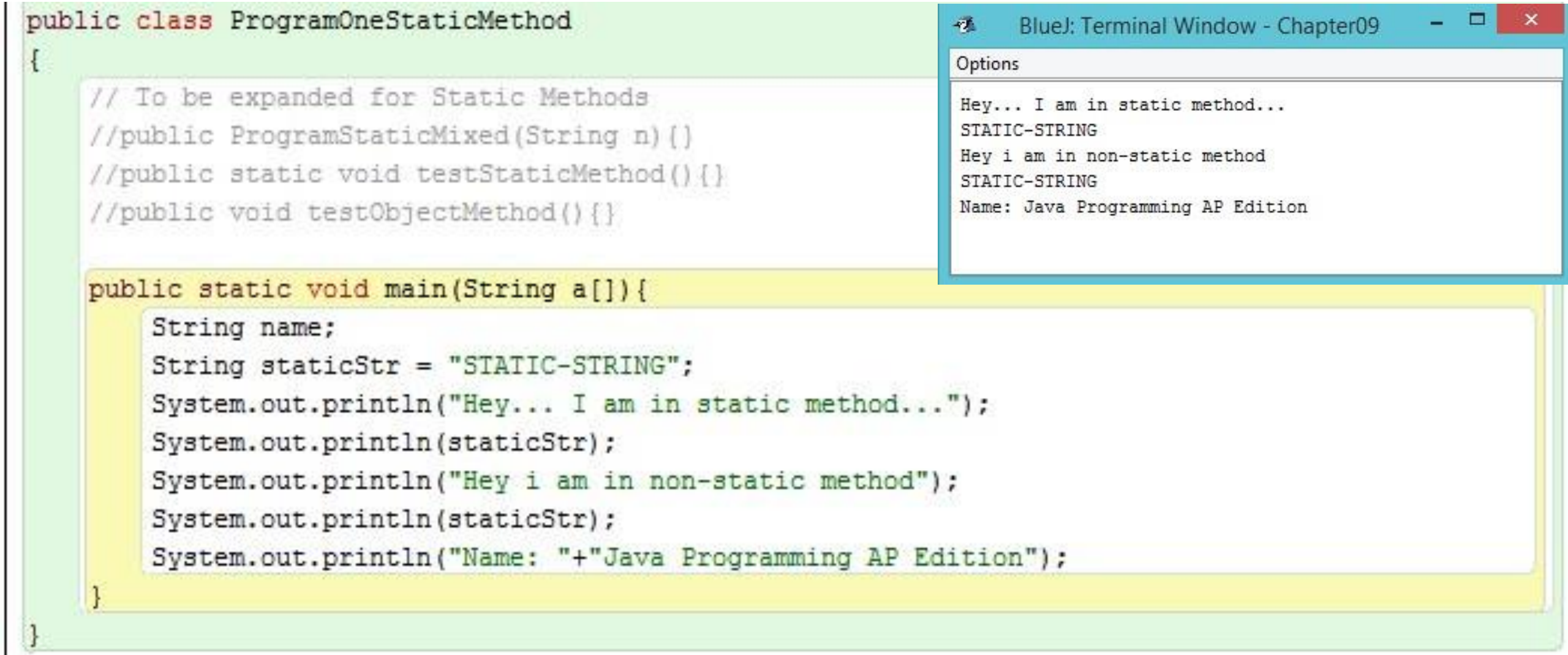
One Static Method Program

ProgramOneStaticMethod.java

- One Method: public static void main(String[] args) {…}
- All variable defined at the beginning of the main method.
- Simple program. No calling other methods. (Elementary Programming: Sequential Programming)

One Static Method Program

ProgramOneStaticMethod.java



```
public class ProgramOneStaticMethod
{
    // To be expanded for Static Methods
    //public ProgramStaticMixed(String n){}
    //public static void testStaticMethod(){}
    //public void testObjectMethod(){}

    public static void main(String a[]){
        String name;
        String staticStr = "STATIC-STRING";
        System.out.println("Hey... I am in static method...");
        System.out.println(staticStr);
        System.out.println("Hey i am in non-static method");
        System.out.println(staticStr);
        System.out.println("Name: "+"Java Programming AP Edition");
    }
}
```

Static Method Only Program Structures

(ProgramStaticOnly.java Equivalent to Structural Programming)

- Global variables defined at the class properties declaration region as static variables.
- Local variables defined at the main method or other methods (or code blocks)
- Use ClassName.method() to call static methods or Use method() to call the methods (if only one class).
- Multiple class, you can only use ClassName.method() to call the method (similar to external functional call for structural programming language like C.)
- Equivalent to Structural Programming like C-language.

Static Method Only Program Structures

(ProgramStaticOnly.java Equivalent to Structural Programming)

```
public class ProgramStaticOnly
{
    private static String staticStr = "STATIC-STRING";

    public static void testStaticMethod(){
        System.out.println("Hey... I am in static method...");
        //you can call static variables here
        System.out.println(ProgramStaticOnly.staticStr);
        //you can not call instance variables here.
    }
    public static void testObjectMethod(String name){
        System.out.println("Hey i am in non-static method");
        //you can also call static variables here
        System.out.println(ProgramStaticOnly.staticStr);
        //you can call instance variables here
        System.out.println("Name: "+name);
    }
    public static void main(String a[]){
        //By using class name, you can call static method
        ProgramStaticOnly.testStaticMethod();
        ProgramStaticOnly.testObjectMethod("Java Programming AP Edition");
    }
}
```

```
BlueJ: Terminal Window - Chapter09
Options
Hey... I am in static method...
STATIC-STRING
Hey i am in non-static method
STATIC-STRING
Name: Java Programming AP Edition
```

Static Method Only with Two Classes

(ProgramStaticOnlyTwo.java Equivalent to Structural Programming)

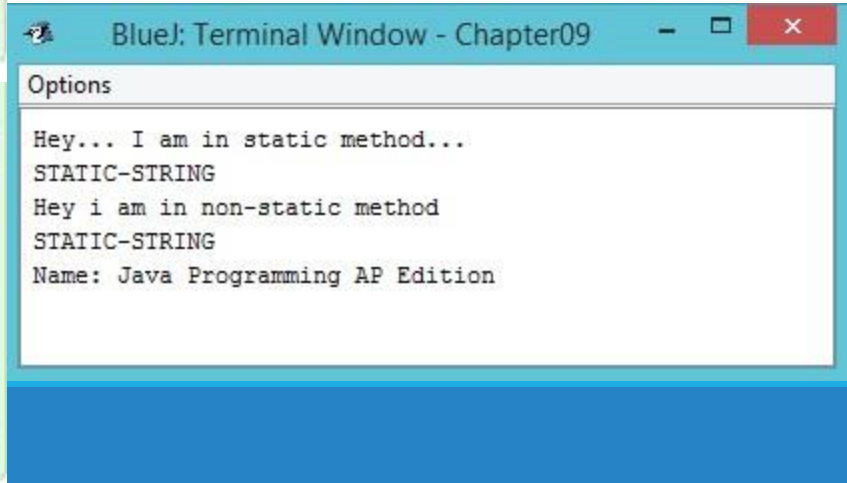
- Similar to Static Method Only Program except that two classes are used. (One is considered as an external program call.
- Because the two classes are in the same package, there is no need to import. If they are in different package, then the visibility modifiers need to be specified as public.
- Similar to external functional call in C-language.
- Math.random() call is considered one of such method calls.
- Two classes are mutually dependent (closely coupled). If they are independent, we call it uncoupled. (closely coupled programs are not considered to be safe.)

Static Method Only with Two Classes

(ProgramStaticOnlyTwo.java Equivalent to Structural Programming)

```
public class ProgramStaticOnlyTwo
{
    public static String staticStr = "STATIC-STRING";
    public static void testStaticMethod(){
        System.out.println("Hey... I am in static method...");
        //you can call static variables here
        System.out.println(ProgramStaticOnlyTwo.staticStr);
        //you can not call instance variables here.
    }
    public static void main(String a[]){
        //By using class name, you can call static method
        ProgramStaticOnlyTwo.testStaticMethod();
        ProgramStaticOnlyTwoSub.testObjectMethod(ProgramStaticOnlyTwoSub.name);
    }
}
```

```
public class ProgramStaticOnlyTwoSub
{   public static String name =  "Java Programming AP Edition";
    public static void testObjectMethod(String name){
        System.out.println("Hey i am in non-static method");
        //you can also call static variables here
        System.out.println(ProgramStaticOnlyTwo.staticStr);
        //you can call instance variables here
        System.out.println("Name: "+name);
    }
}
```



Program Structure Static Mixed Style

Program: Example for static variables and methods

In java, static belongs to class. You can create static variables and static methods. You can call these directly by using class name, without creating instance.

Java static variables:

Static variables are belongs to the class and not to the object. These are only once, at the starting of the execution. Static variables are not part of object state, means there is only one copy of the values will be served to all instances. You can call static variable with reference to class name without creating an object. Below example shows how to create and call static variables.

Java static methods:

Static methods are also similar to static variables, you can access them with reference to class name, without creating object. Inside static methods, you cannot access instance variables or instance methods. You can only access static variables or static methods.

Program Structure Static Mixed Style

ProgramStaticMixed.java

```
public class ProgramStaticMixed {
    private String name;
    private static String staticStr = "STATIC-STRING";
    public ProgramStaticMixed(String n){
        this.name = n;
    }
    public static void testStaticMethod(){
        System.out.println("Hey... I am in static method...");
        //you can call static variables here
        System.out.println(ProgramStaticMixed.staticStr);
        //you can not call instance variables here.
    }
    public void testObjectMethod(){
        System.out.println("Hey i am in non-static method");
        //you can also call static variables here
        System.out.println(ProgramStaticMixed.staticStr);
        //you can call instance variables here
        System.out.println("Name: "+this.name);
    }
    public static void main(String a[]){
        //By using class name, you can call static method
        ProgramStaticMixed.testStaticMethod();
        ProgramStaticMixed msm = new ProgramStaticMixed("Java Programming AP Edition");
        msm.testObjectMethod();
    }
}
```

```
BlueJ: Terminal Window - Chapter09
Options
Hey... I am in static method...
STATIC-STRING
Hey i am in non-static method
STATIC-STRING
Name: Java Programming AP Edition
```

Data-Centric Program (OOP)

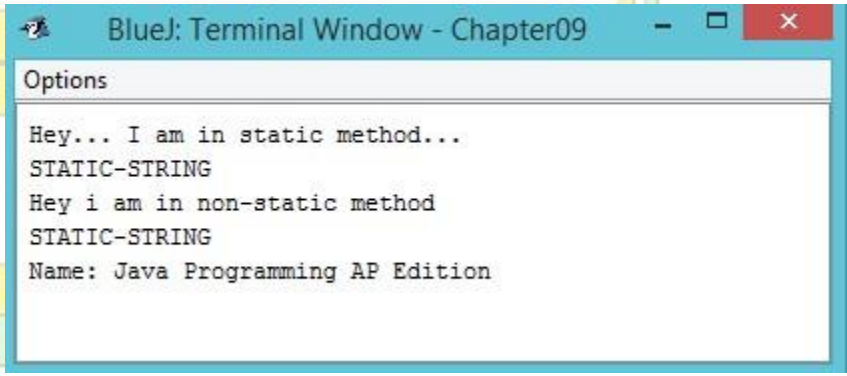
ProgramObjectOriented.java


```
public class ProgramOOP
{   private String name;
    private String staticStr;
    // It is not a static String, just make it look similar to show the different program style.
    ProgramOOP(){}
    ProgramOOP(String n, String str){
        name = n;
        staticStr = str;
    }
    public String getName() {return name; }
    public String getStr()  {return staticStr; }
    public void setName(String n) {name = n;}
    public void setStr(String str){staticStr = str;}
}
```

Tester Class for Data-Centric Program

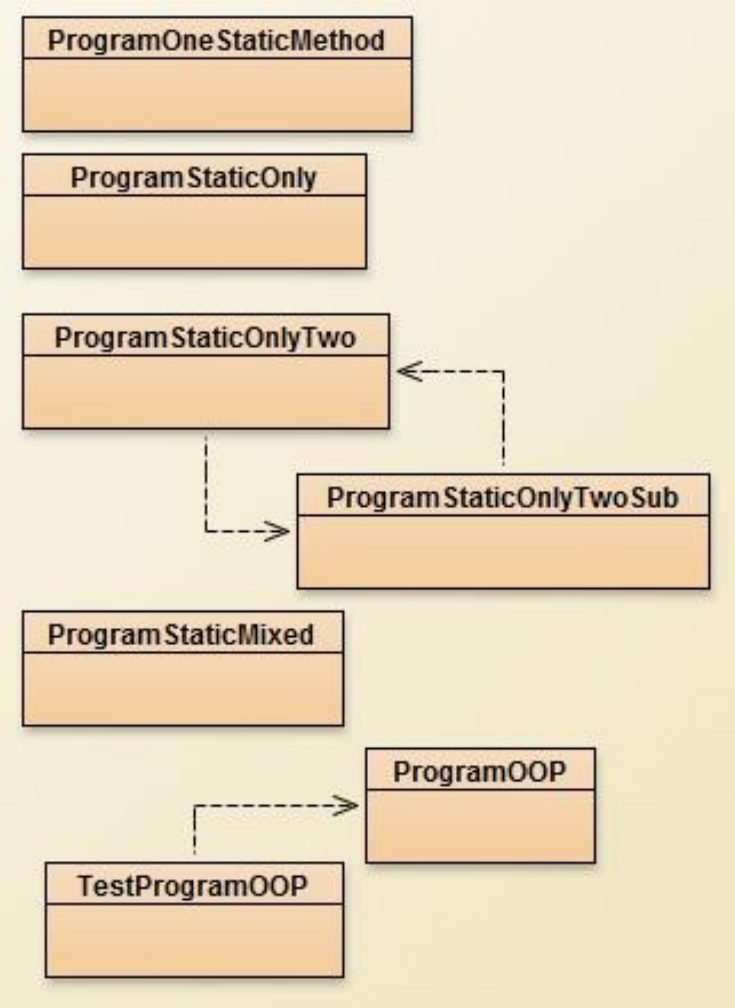
TestObject-Oriented.java

```
public class TestProgramOOP
{   public static void testStaticMethod(ProgramOOP oop){
        System.out.println("Hey... I am in static method...");
        oop.setStr("STATIC-STRING");
        System.out.println(oop.getStr());
    }
    public static void testObjectMethod(ProgramOOP oop){
        System.out.println("Hey i am in non-static method");
        System.out.println(oop.getStr());
        oop.setName("Java Programming AP Edition");
        System.out.println("Name: "+oop.getName());
    }
    public static void main(String[] args){
       ProgramOOP oop = new ProgramOOP();
       testStaticMethod(oop);
       testObjectMethod(oop);
    }
}
```



Download the Zip file and Try on These Programs

(ClassAndObjectProgram.zip)



For the same method names, variable name, and the output, we have seen them in different program structures.

In this lecture, I just try to give audience a picture of the many program structures that he might choose to use in Java. Some simple, some more complicated. The contents and the output really do not matter that much.