POLBAN

Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

Pertemuan 4 Object, Class, Encapsulation

D4 Kelas 1A/1B

Dosen Pengampu :
Zulkifli Arsyad, Wendy Wirasta

Topics

- Object & Class
- Instance Fields
- Constructor
- Encapsulation

Object & Class

- Kelas adalah template atau blueprint dari mana objek dibuat
- Object adalah instance dari class
- When you construct an object from a class, you are said to have created an instance of the class
- Instance Fields is variable

```
public class EmployeeTest
{
   public static void main(String[] args)
   {
      // fill the staff array with three Employee objects
      Employee[] staff = new Employee[3];

      staff[0] = new Employee("Carl Cracker", 75000, 1987, 12, 15);
      staff[1] = new Employee("Harry Hacker", 50000, 1989, 10, 1);
      staff[2] = new Employee("Tony Tester", 40000, 1990, 3, 15);
```

```
class Employee
{
   // instance fields
   private String name;
   private double salary;
   private LocalDate hireDay;

   // constructor
   public Employee(String n, double s, int year, int month, int day)
   {
      name = n;
      salary = s;
      hireDay = LocalDate.of(year, month, day);
   }

   // a method
   public String getName()
   {
      return name;
   }

   // more methods
   . . .
}
```

Constructor

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

Encapsulation

- hiding the implementation details / wrapping the data (variables)
- methods never directly access instance fields in a class other than their own. Programs should interact with object data only through the object’s methods.

- To achieve encapsulation in Java
  - Declare the variables of a class as private.
  - Provide public setter and getter methods to modify and view the variables values (accessor & Mutator Method)

```java
/* File name : EncapTest.java */
public class EncapTest {
   private String name;
   private String idNum;
   private int age;

   public int getAge() {
      return age;
   }

   public String getName() {
      return name;
   }

   public String getIdNum() {
      return idNum;
   }

   public void setAge( int newAge) {
      age = newAge;
   }

   public void setName(String newName) {
      name = newName;
   }

   public void setIdNum( String newId) {
      idNum = newId;
   }
```

Access Modifier

Java offers four choices of access modifier:

public The method can be called from any class.

private The method can only be called from within the same class.

protected The method can only be called from classes in the same package or subclasses. You'll learn about subclasses in Chapter 5.

Default (Package Private) Access The method can only be called from classes in the same package. This one is tricky because there is no keyword for default access. You simply omit the access modifier.