

Classes and Objects

Accessibility of methods

- The **access control modifier** (`public`, `private`) at the beginning of the method declaration defines where the method can be accessed from.
- `public` – methods can be seen and called from anywhere, from any method in any class in any package. The `main` method must be `public`.
- `private` – methods can only be accessed from within the class in which it is defined.
- Omission of access control modifier – methods can be accessed from within the package in which it is defined.

Constructors

```
public Fraction (int n, int d) {  
    num = n;  
    den = d;  
}
```

- A *constructor* is a method that has the same name as the class
- A *constructor* does not specify a return type (not even `void`). It is implicit that the method returns an instance of the class.
- It is called with the keyword `new` to create an instance of the object, e.g.,

```
Fraction f = new Fraction();
```
- *Constructors* are used to initialize objects at the time the objects are created
- A default constructor is automatically created by Java, it initializes all numeric fields to zero, `boolean` fields to `false`, and reference fields to `null`
- You can tailor your initialization by writing your own constructors
- There can be more than one constructor, but they must have different parameter list (to differentiate them)

Hiding Information

- The **visibility modifier** (`public`, `private`) before the field declaration defines where the field is available to be read and altered from.
- `public` – field is openly available to be read and altered from within any class in any package
- Omission of visibility modifier – field is available for inspection and alteration from within any class in the package in which the class is defined.
- `private` – field is only available to be read and altered within the class that the field is defined.
- Fields should often be made `private` so the class has better control of them because other classes can not have direct access.
- In order to access private fields from other classes, accessor and mutator methods are required.

```
public int getNumerator() {  
    return num;  
}  
  
public void setNumerator(int n) {  
    num = n;  
}
```