The Object-Oriented Thought Process

Chapter 04

The Anatomy of a Class

Contents

- The Name of the Class
- Comments
- Attributes
- Constructors
- Accessors
- Public Interface Methods
- Private Implementation Methods

The Name of the Class

The name of the class is important for several reasons.

- To identify the class itself.
- The name must be descriptive.
- The choice of a name is important because it provides information about what the class does and how it interacts within larger systems.

Simple Class (1/3)

```
/*
  This class defines a cabbie and assigns a cab
*/
public class Cabbie{

  //Place name of Company Here
  private static String companyName = "Blue Cab Company";

  //.Name of the Cabbie
  private String Name;

  //Car assigned to Cabbie
  private Cab myCab;
```

Class Name

Attributes

Simple Class (2/3)

Constructors

```
// Default Constructor for the Cabbie
public Cabbie() {

   Name= null;
   myCab = null;

}

// Name Initializing Constructor for the Cabbie
public Cabbie(String iName, String serialNumber){

   Name = iName;
   myCab = new Cab(serialNumber);
}
```

Simple Class (3/3)

```
  // Set the Name of the Cabbie
  public void setName(String iName) {
     Name = iName;
  }

// Get the Name of the Company
public        string getName(){
  return Name;
}

// Get the Name of the Cabbie
  public static String getCompanyName(){
     return companyName;
  }

  public void giveDestination(){
  }

  private void turnRight(){
  }

  private void turnLeft(){
  }

}
```

Accessor Methods

A Public Interface 

Private Implementation

Comments

Regardless of the syntax of the comments used, comments are vital to understanding the function of a class.

- While comments are vital to the documentation and maintenance of code, it is important not to over-comment.

```
/*
  multi-line comments
*/
// single line comments
```

Attributes

Attributes represent the state of the object because they store the information about the object.

- In many designs all attributes are private.
- Keeping the interface design as minimal as possible.
- The only way to access these attributes is through the method interfaces provided.

- The keyword private signifies that a method or variable can be accessed only within the declaring object.
- The static keyword signifies that there will be only one copy of this attribute for all the objects instantiated by this class.
  - C.f., Class attributes

- E.g., Class attributes

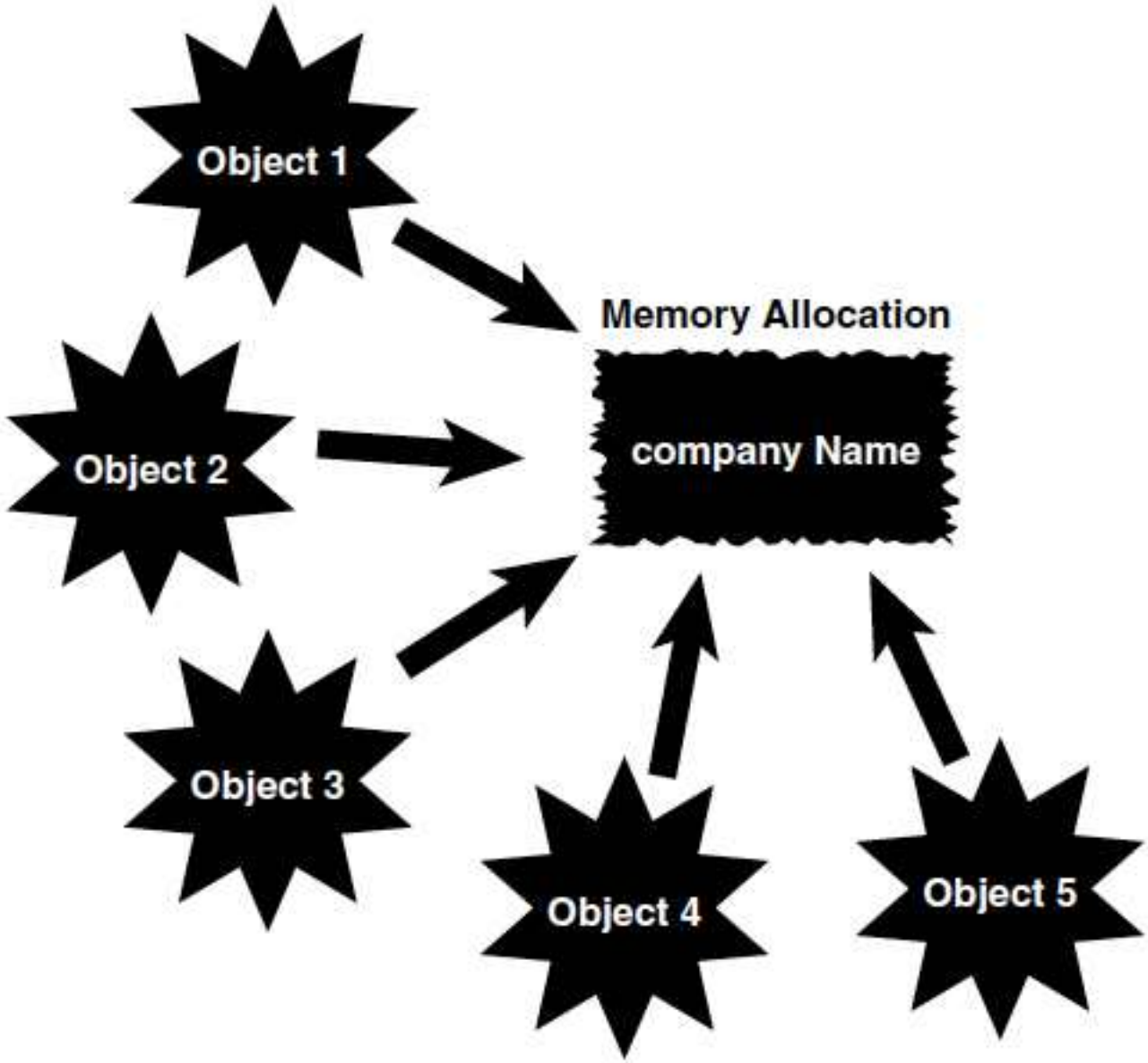

Figure 4.2 Object memory allocation.

Methods

Methods represent the behavior of the object because they provide the functionality.

- Methods are defined by their signature and are used to invoke certain behaviors.
- One of the uses of methods is as accessor methods.

Constructors

```
public Cabbie(String iName, String serialNumber) {
        Name = iName;
        myCab = new Cab(serialNumber);
}
```

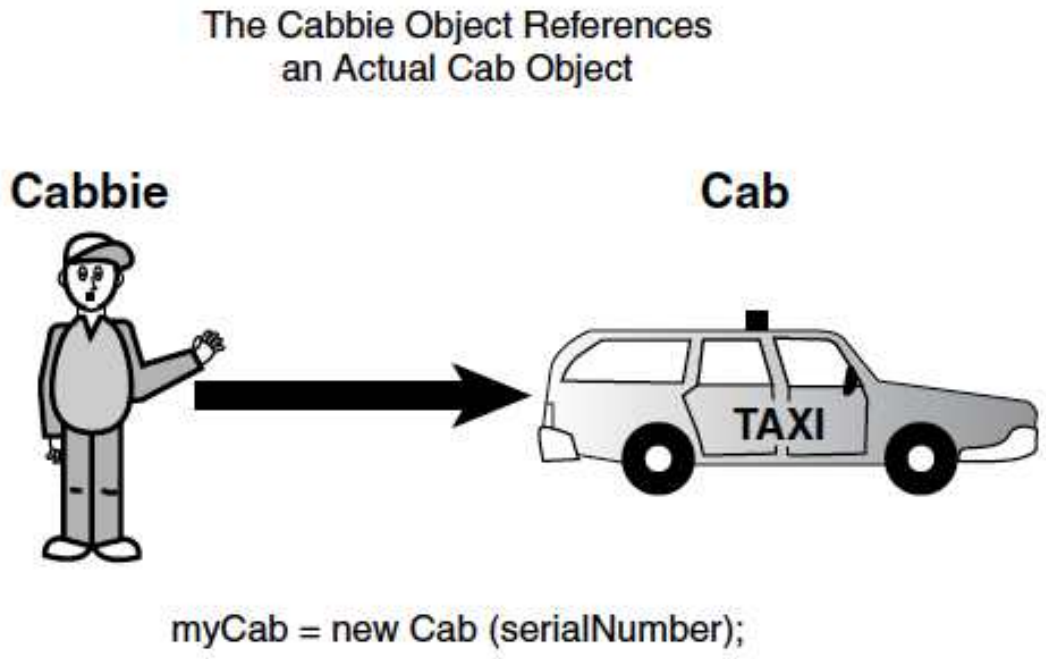

Figure 4.3 The **`Cabbie`** object referencing an actual cab object.

Accessor Methods

Controlled access to attributes is provided by methods.

- These method are called *accessors.*
- Sometimes accessors are referred to as getters and setters.

```
public void setName(String iName) {
    Name = iName;
}
public String getName() {
    return Name;
}
```

static method

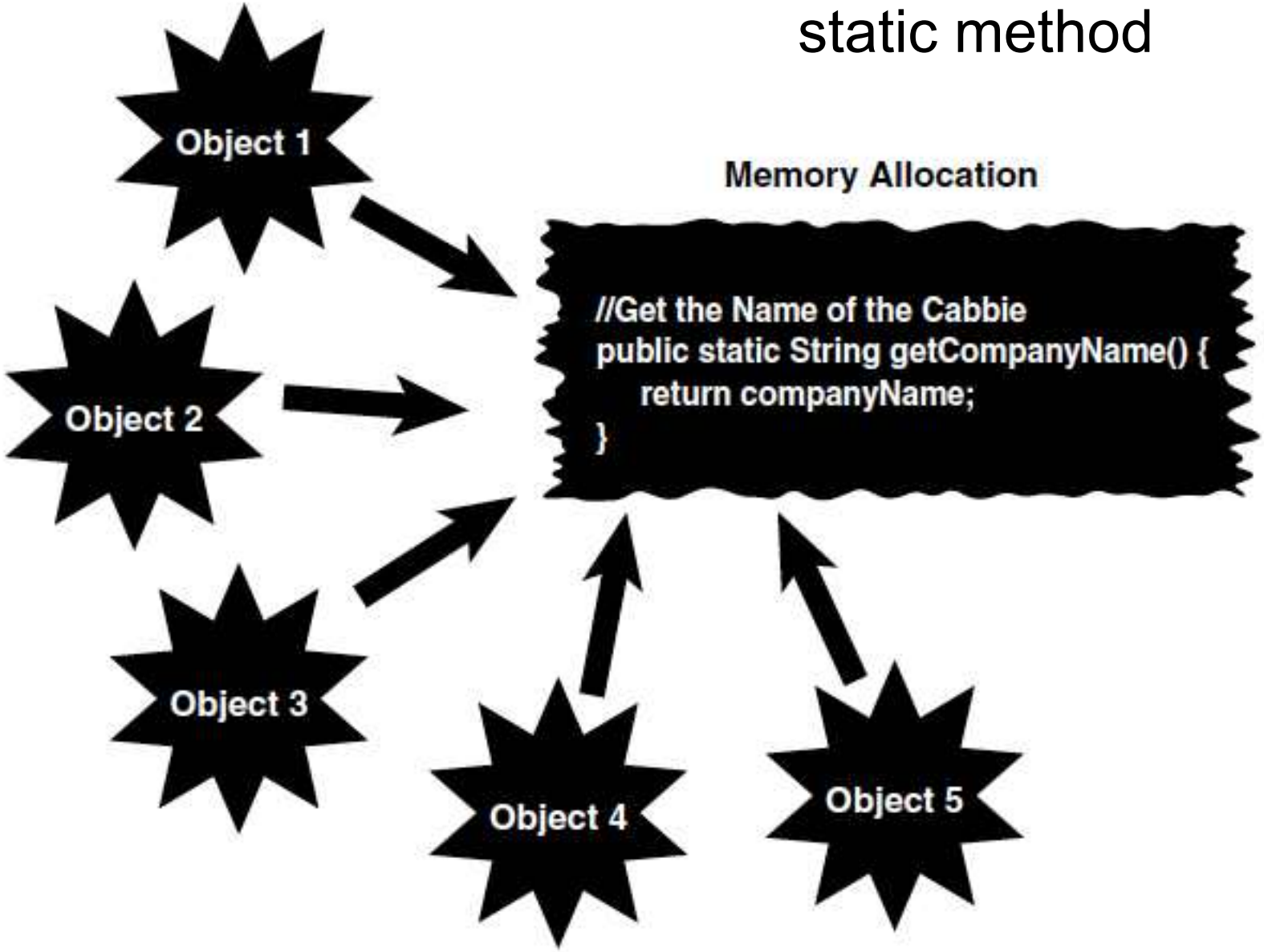

Figure 4.5 Method memory allocation.

Public Methods

Both the constructors and the accessor methods are declared as public and are part of the public interface.

- Other methods can be part of the public interface as well.

```
public void giveDestination (){
}
```

Private Methods

It is common for methods in a class to be hidden from other classes. These methods are declared as private:

```
private void turnRight(){
}
private void turnLeft() {
}
```

- Private methods could be called from within the method giveDestination :

```
public void giveDestination (){
    .. some code
    turnRight();
    turnLeft();
    .. some more code
}
```