CSCI3170 Introduction to Database Systems

Tutorial 1 – Introduction to Java (I)

Outline

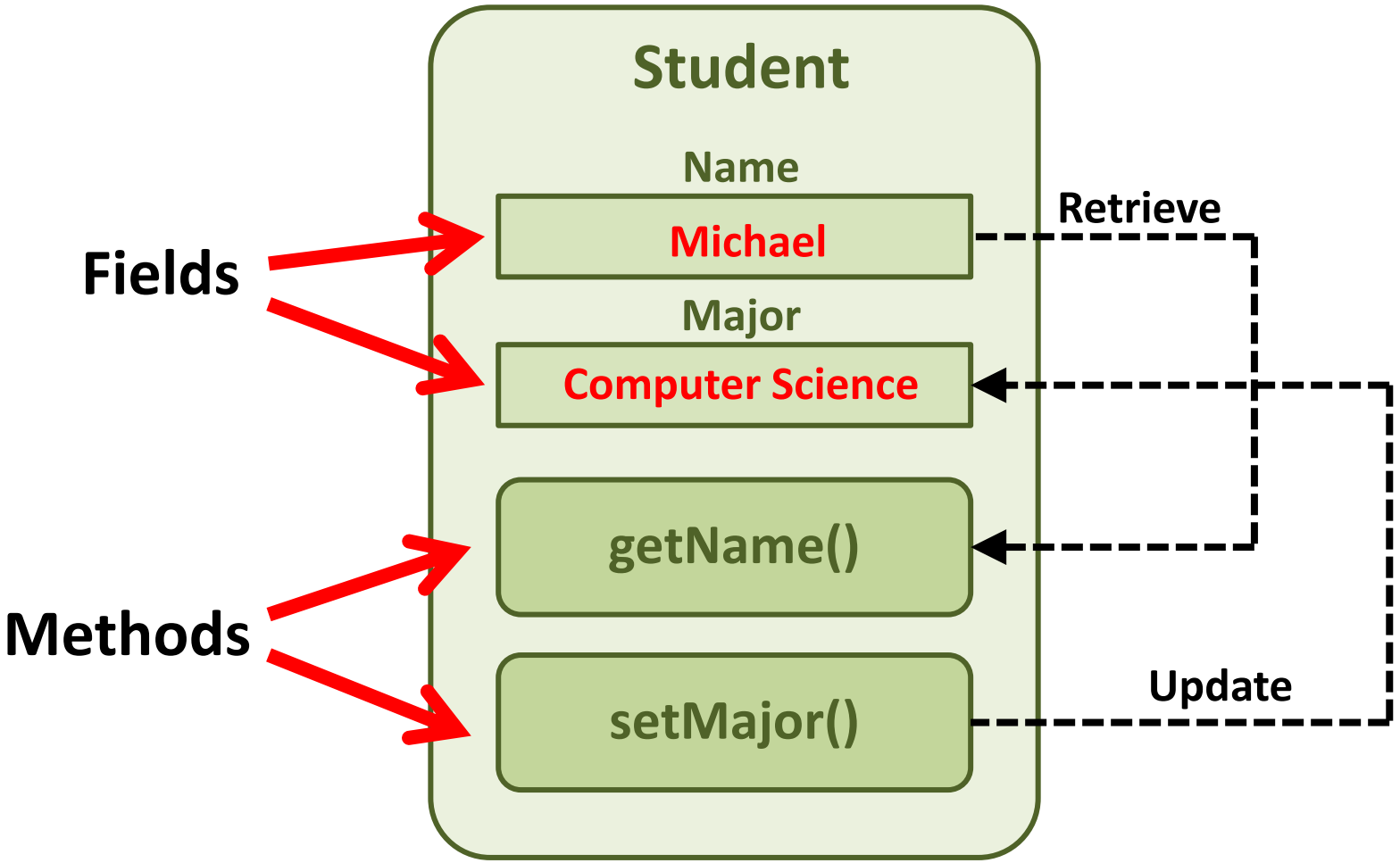
- Object-Oriented Programming
- Basic Java
- Compile and Run
- Useful Reference Sites

OBJECT-ORIENTED PROGRAMMING

Object

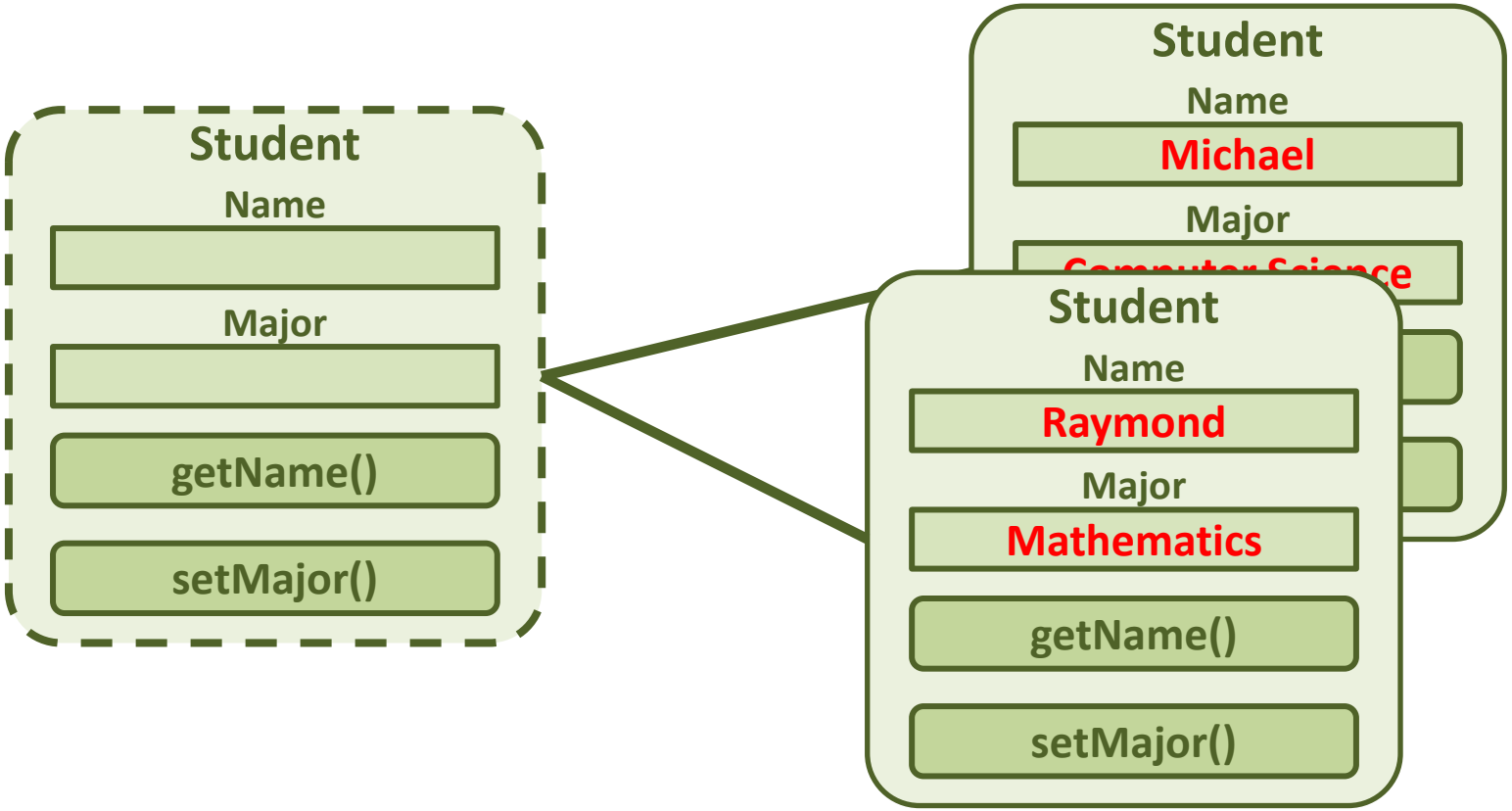
- Objects are representations of real world entities
- Objects have two properties
  - **Field**
  - **Method**
- Fields tell you what an object is (properties)
- Methods tell you what an object does (tasks)

Field and Method



Class

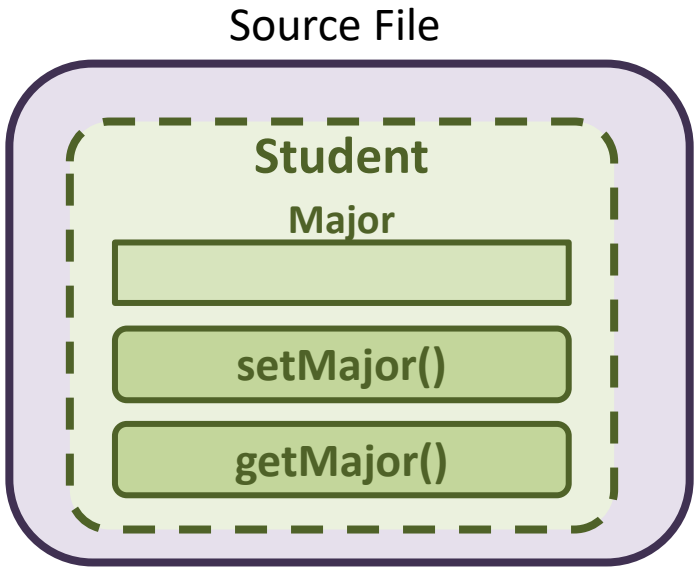
- A class is a template to create objects
- A class is often defined by a source file



Object Reference

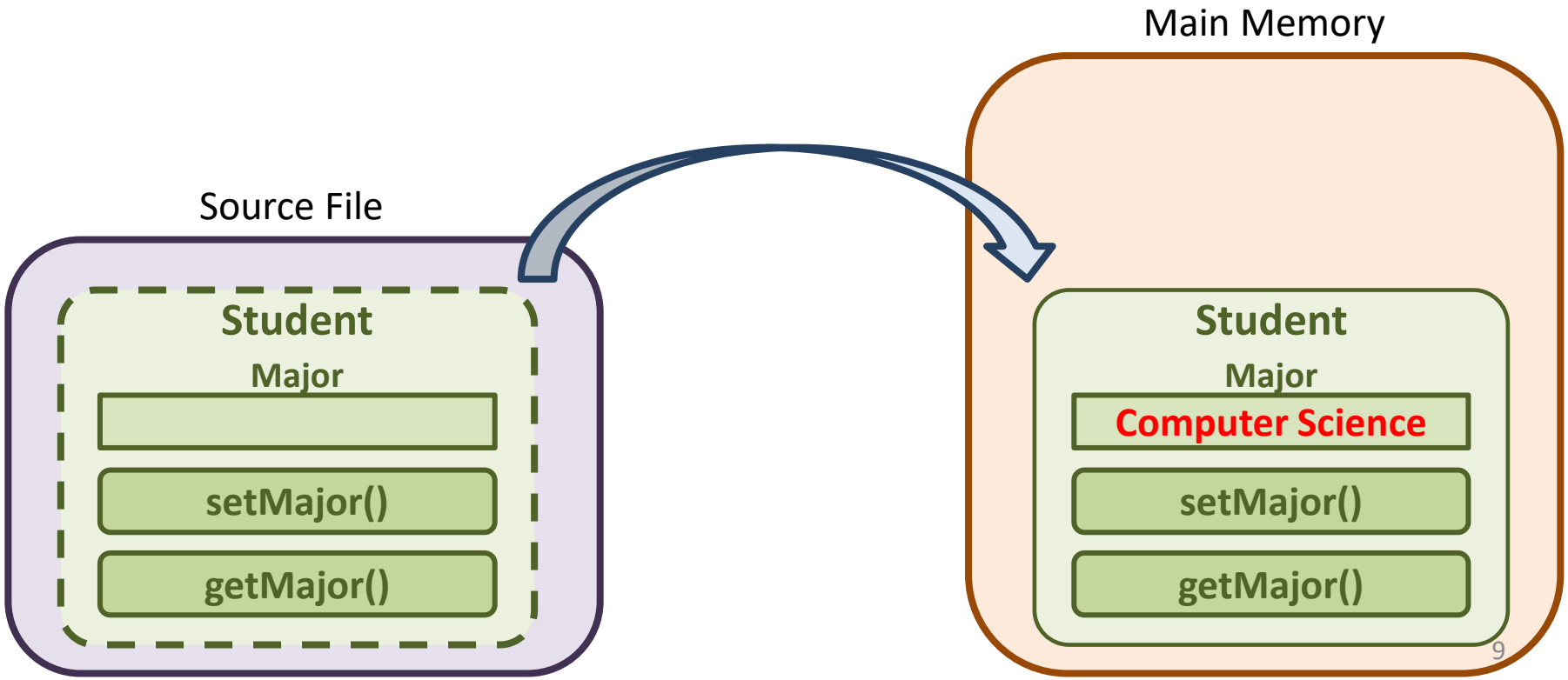
- When a object is created, it is stored in main memory.
- To access a object, we need to know its address
- **Object Reference** is a variable which stores the memory address of an Object

Example



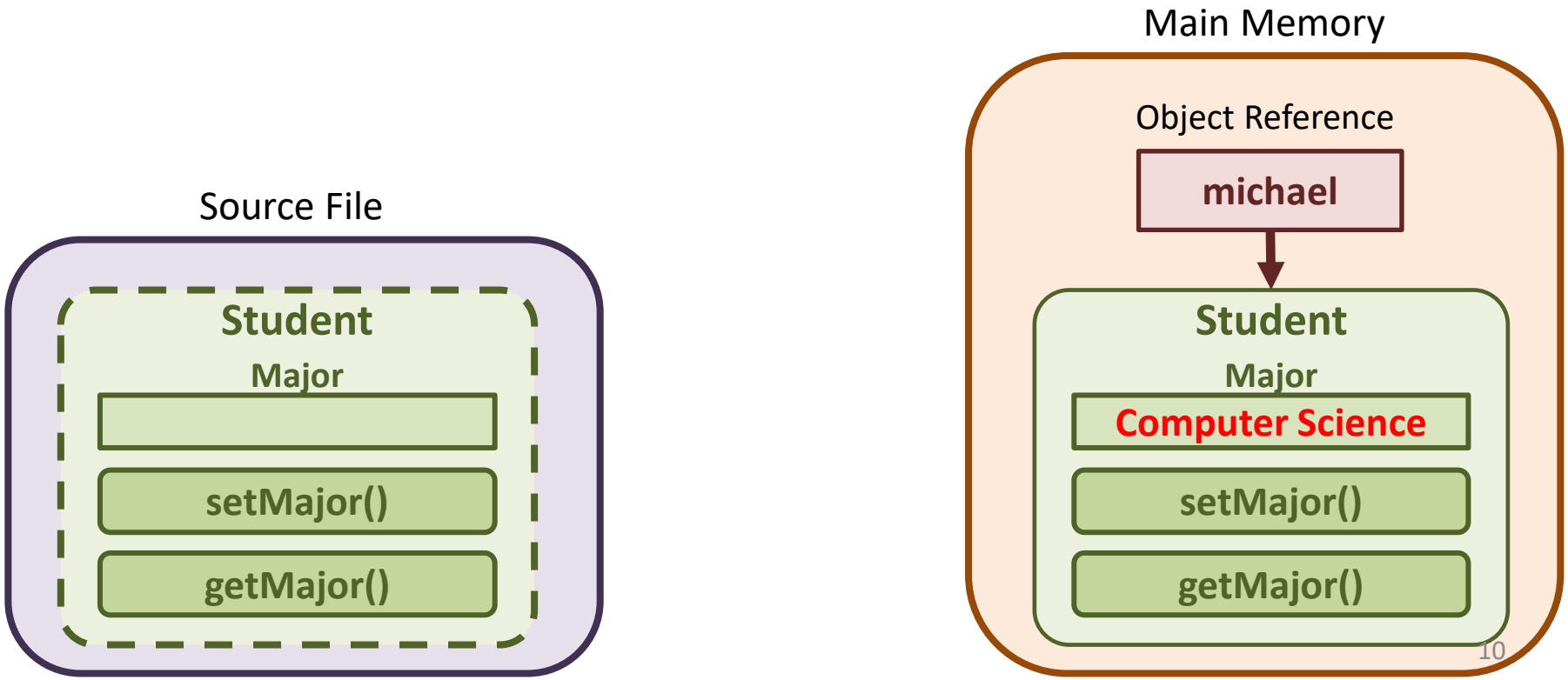
Example

A Object is created in main memory according to a class defined in source file



Example

An object reference is created such that we can access the object later through that object reference



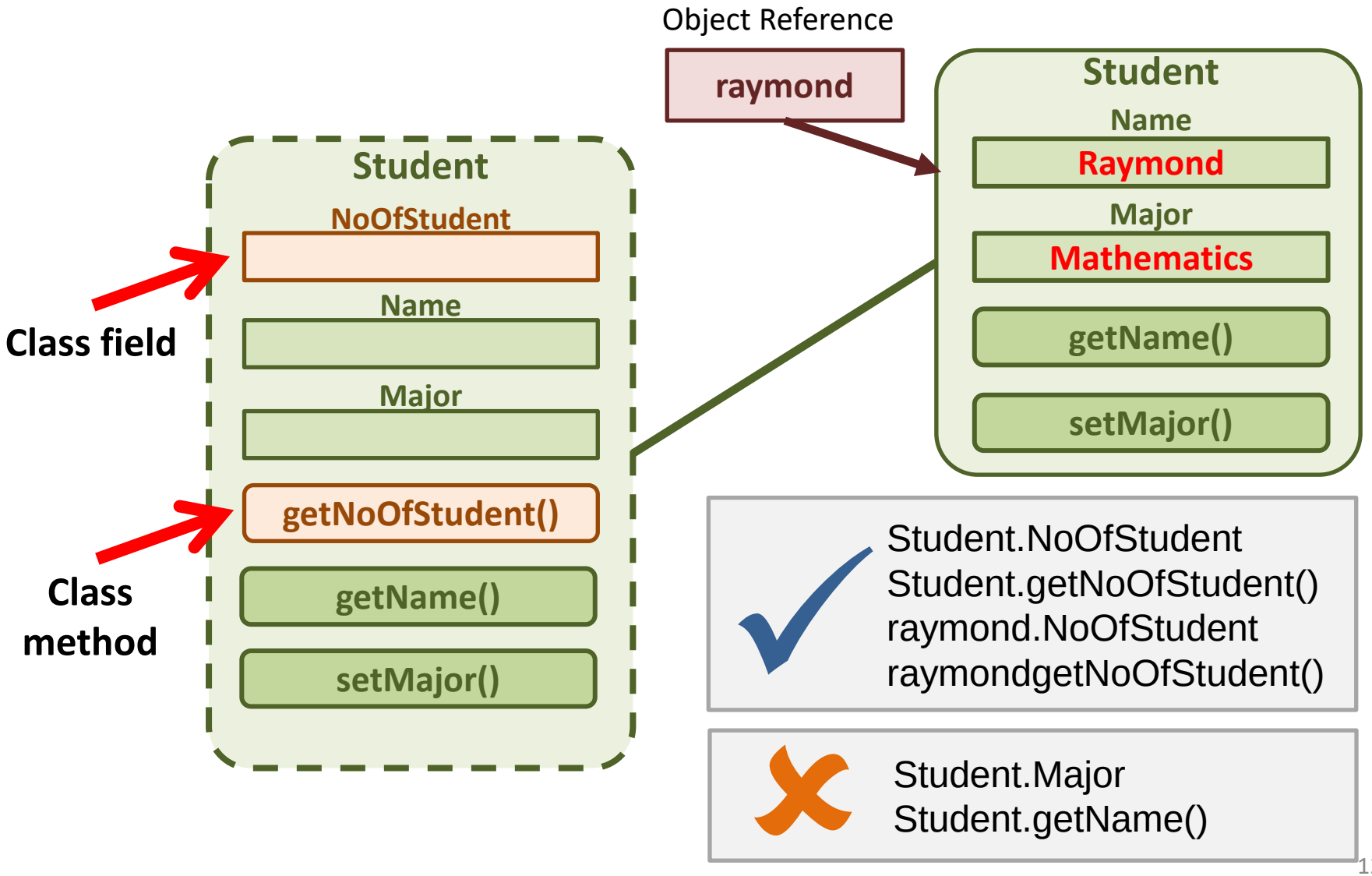
Class field and Class method

- A class can also have fields and methods!

Class field and class method belong to class and can be accessed without an object.

Object field and object method belong to an object and can be accessed by that object through its objective reference

Class field and Class method (2)



Syntax and API

BASIC JAVA

First Program - Hello World

```
//HelloWorld.java
import java.io.*;

class HelloWorld{
    public static String mesg = "Hello World!";
    public static void main(String[] args){
        System.out.println(mesg);
    }
}
```

First Program - Hello World

```
//HelloWorld.java
import java.io.*;

class HelloWorld{
    public static String mesg = "Hello World!";
    public static void main(String[] args){
        System.out.println(mesg);
    }
}
```

A Class

- It stores the object constructors, class methods and object methods

First Program - Hello World (2)

```
//HelloWorld.java
import java.io.*;

class HelloWorld{
    public static String mesg = "Hello World!";
    public static void main(String[] args){
        System.out.println(mesg);
    }
}
```

A Field

- It store the properties of the object or class
  - **Class Field**: Field belongs to a class
  - **Object Field**: Field belong to a specific object

First Program - Hello World (3)

```
//HelloWorld.java
import java.io.*;

class HelloWorld {
    public static String mesg = "Hello World";
    public static void main(String[] args){
        System.out.println(mesg);
    }
}
```

A Method

- It is a segment of codes that can perform a specific task
  - **Constructor**: Creates a object
  - **Class Method**: Methods belongs to a class
  - **Object Method**: Methods belong to a specific object

First Program - Hello World (4)

```
//HelloWorld.java
import java.io.*;

class HelloWorld {
    public static String mesg = "Hello World!";
    public static void main(String[] args){
        System.out.println(mesg);
    }
}
```

A Statement

- A command that the computer can execute
- **Ends with a semicolon!!**

First Program - Hello World (5)

```
//HelloWorld.java
import java.io.*;

class HelloWorld {
    public static String mesg = "Hello World!";
    public static void main(String[] args) {
        System.out.println(mesg);
    }
}
```

A Comment Statement

- A remark on the implementation details of the program
- A good programmer should always write comments

First Program - Hello World (6)

```
//HelloWorld.java
import java.io.*;

class HelloWorld {
    public static String mesg = "Hello World!";
    public static void main(String[] args) {
        System.out.println(mesg);
    }
}
```

A Import Statement

- Tells the compiler to find for external library during compilation

Data type in Java

- The value of field/local variables are restricted by their **data type**.
- 2 major kinds of data type are:
  - 8 Primitive Type
    - Number (byte, short, int, long, float, double)
    - Character (char)
    - Boolean (boolean)
  - Object Type
    - Store a specific kind of object

Primitive Data type in Java

Type	Description
byte	8 bit integer range: [-128, 127]
short	16 bit integer range: [-32768, 32767]
int	32 bit integer range: $[-2^{31}, 2^{31}-1]$
long	64 bit integer range: $[-2^{64}, 2^{64}-1]$
float	32-bit IEEE 754 floating point number
double	64-bit IEEE 754 floating point number
char	16-bit Unicode character
boolean	True/False Boolean Value

Conditional Statement(1)

```
if(<expression 1>){
    <statement(s)>
}else if(<expression 2>){
    <statement(s)>
}else{
    <statement(s)>
}
```

- Evaluate expression are from top to bottom

- An expression is true →
  - Its corresponding statement would be executed.
  - All expressions at the bottom would be ignored

- No expression is true→
  - Statements at else section is executed

Conditional Statement (2)

```
switch(<variable>){
    case <value>:   <statement(s)>
                    break;
    case <value>:   <statement(s)>
                    break;
    default: <statement(s)>
}
```

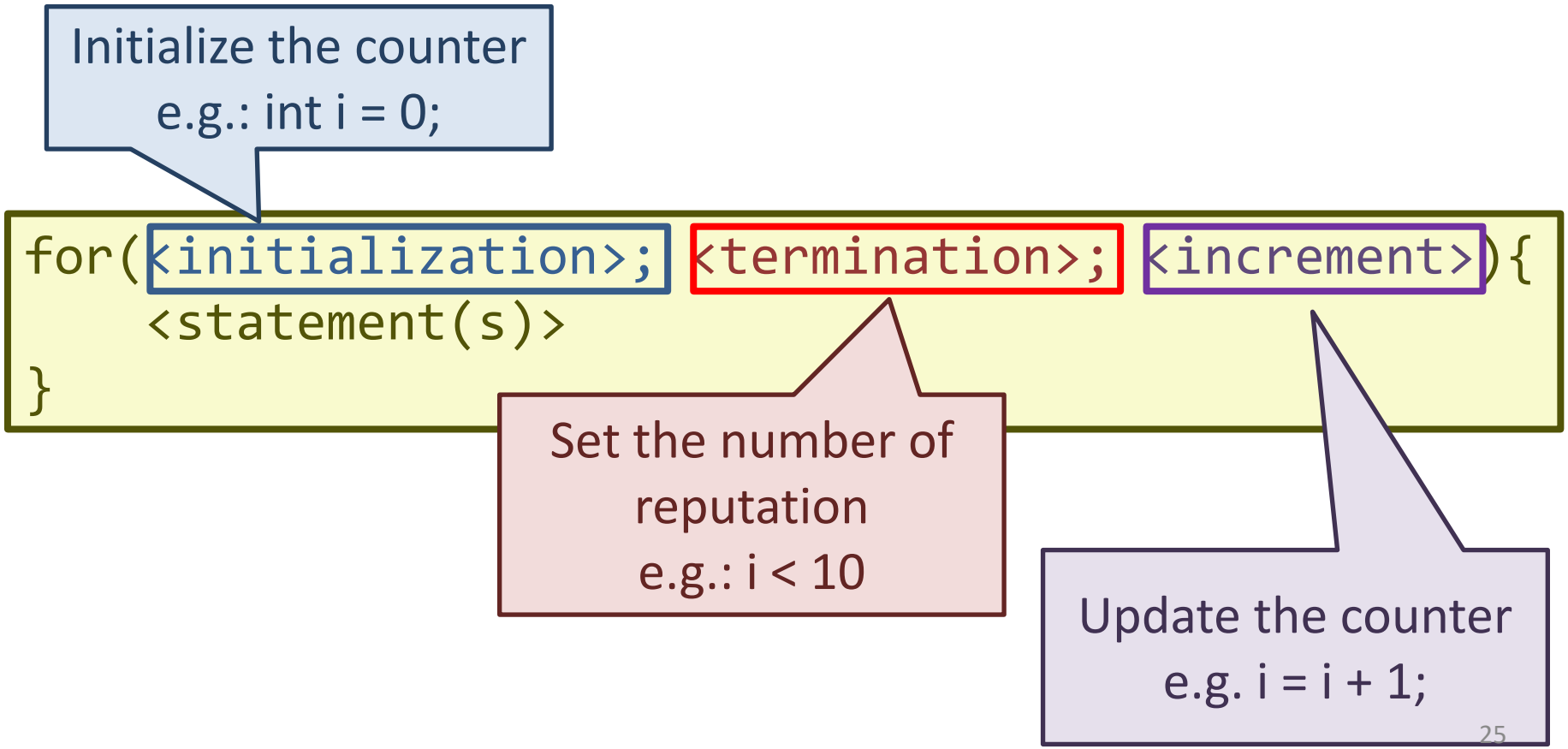
- Compare variable with conditional values from top to bottom

- No match→
  - Statements at default section is executed

- If there is a match →
  - Its corresponding statement would be executed.
  - All values at the bottom would be ignored

For Loop

Repeat the execution of a sequence of statements for a specific number of time



While and Do-While Loop

Repeat the execution of a sequence of statements until the loop citation is validated

While Loop:

```
while(<expression>){
    <statement(s)>
}
```

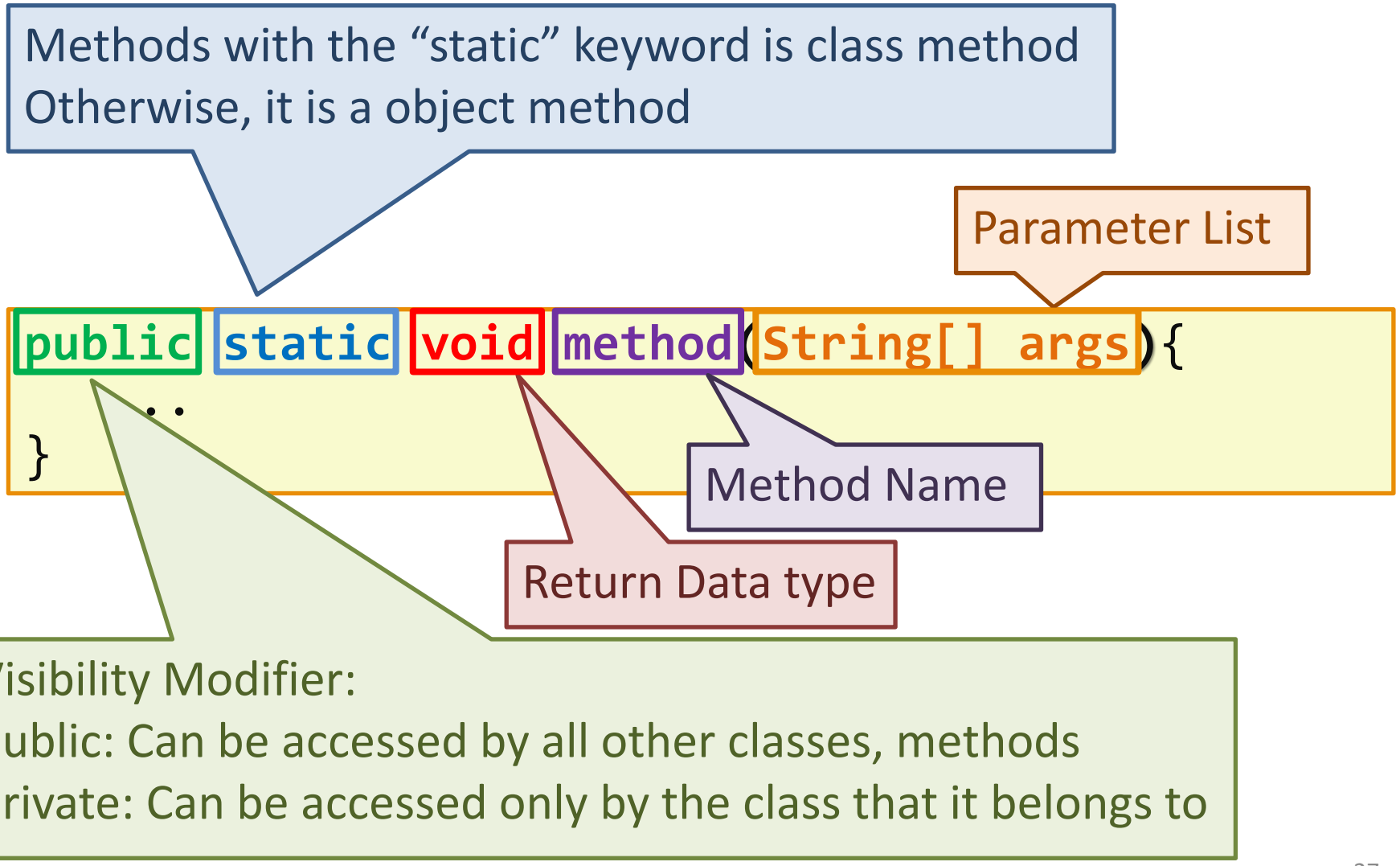
Do-while Loop:

```
do{
    <statement(s)>
} while(<expression>);
```

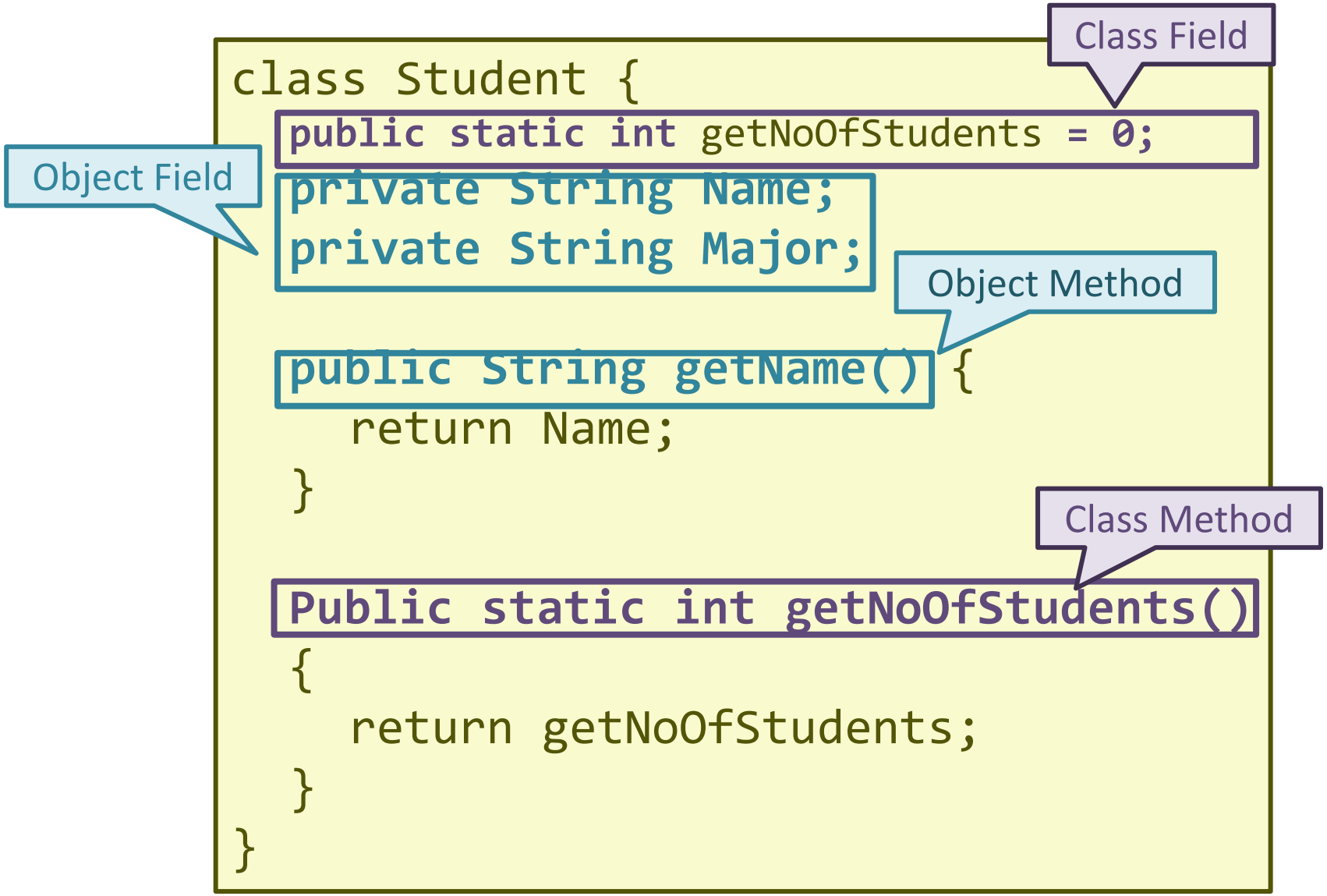
The loop would continue iff the expression (loop citation) is true

The loop citation check of do-while loop is done at the bottom → The statement being repeated are ensure to be executed once!

Define a Method



Class Source File



Create an Object

```
//HelloWorld.java
import java.io.*;

class HelloWorld {

    public static void main(String[] args) {
        Student alice, james, rose;
        alice = new Student("Alice", "CEG");
    }
}
```

Declaring an Object reference Variable

Assigning address of the object created to object reference variable

Creating a new object according to a classs

Unix and Windows configurations

COMPILE AND RUN

Compile and Run

- To run a Java program:

1. Compile the source code file as byte code program through Java complier (**javac**)

```
Example: javac HelloWorld.java
```

2. Run the byte code program through Java interpreter (**java**)

```
Example: java HelloWorld
```
(No ".java" / ".class" at the end)

Useful Reference Sites

- The Java Tutorial
  - http://download.oracle.com/javase/tutorial/
- Java APIs
  - http://download.oracle.com/javase/6/docs/apii/
- CSE Summer Preparatory Course
  - http://www.cse.cuhk.edu.hk/~csesc/
- Free Electronic Book: Thinking in Java, 3rd Edition
  - http://www.mindviewinc.com/Books/downloads.html