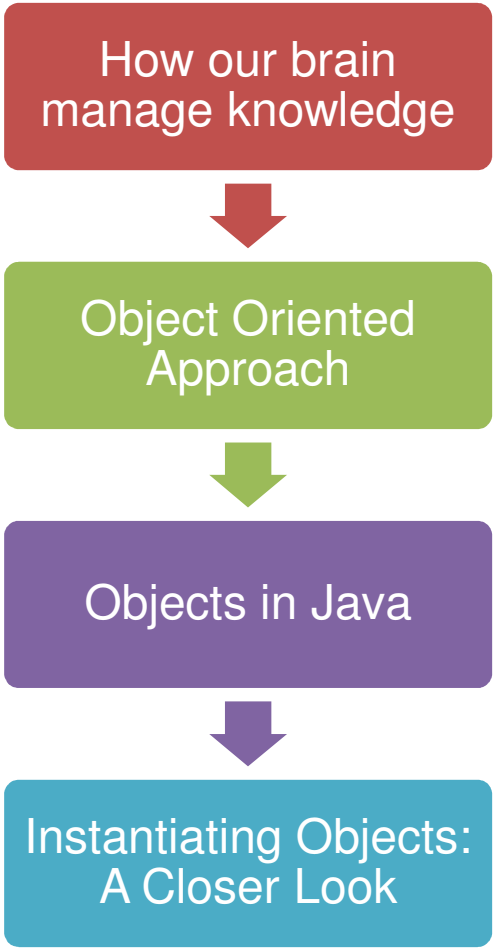
Objects & Classes I

Christian Rodríguez Bustos

Object Oriented Programming

Agenda


How our brain manage knowledge
Object Oriented Approach
Objects in Java
Instantiating Objects: A Closer Look

How our brain manage knowledge

Abstraction

Abstraction hierarchy

Abstraction and software development

How our brain manage knowledge

How our brain manage knowledge

What do you remember?

How our brain manage knowledge

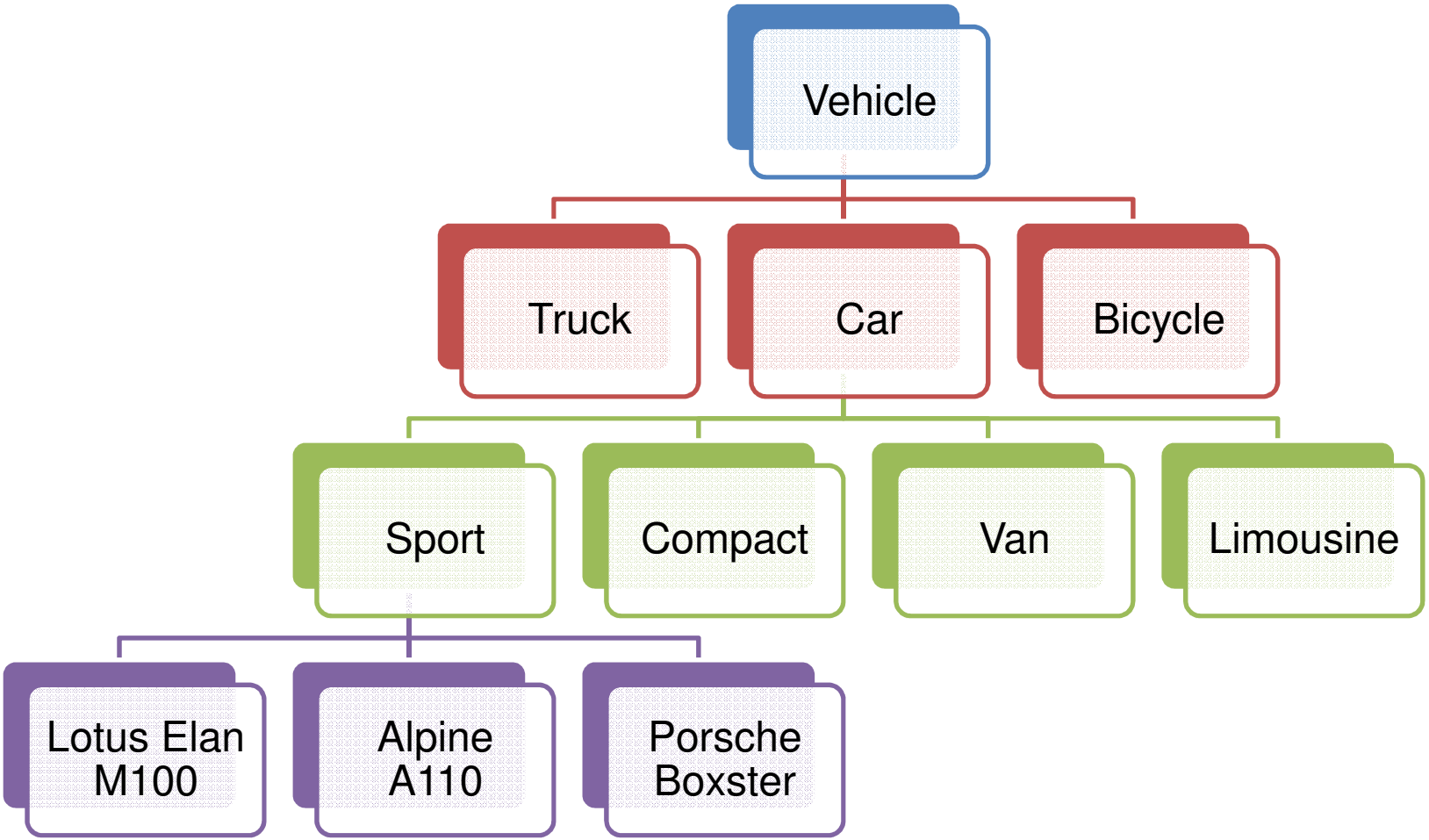
Our brains naturally **simplify** the details of all that we observe.

Details are manageable through a process known as **abstraction**.

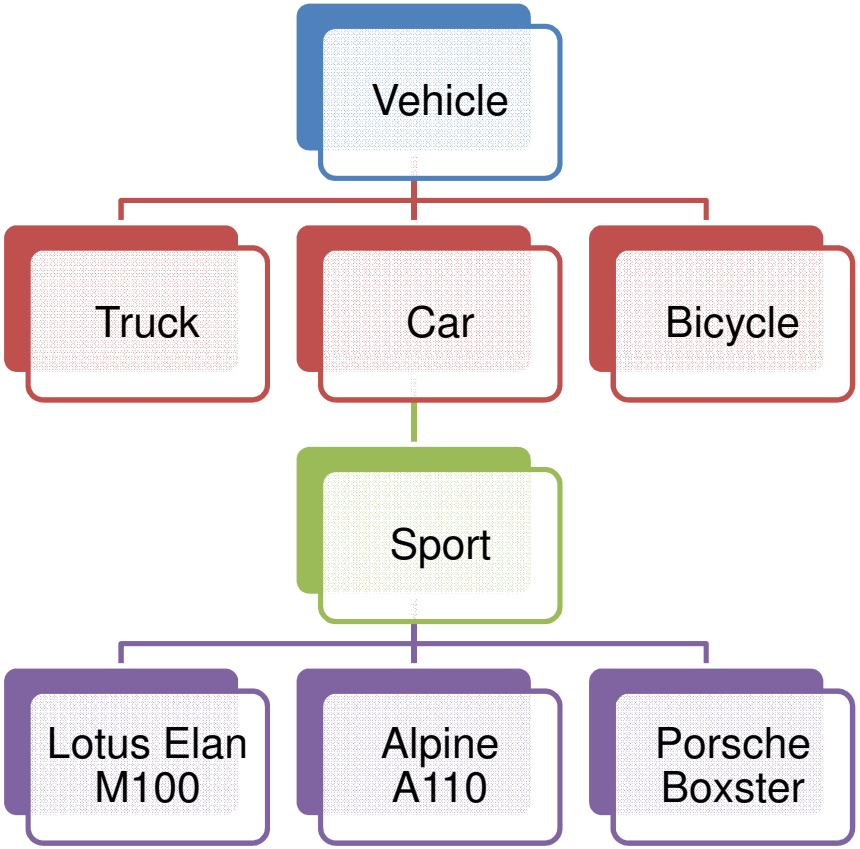
Abstraction

Process that involves **recognizing and focusing** on the **important characteristics of a situation or object**, and filtering out or ignoring all of the **unessential details**.

Simple abstraction hierarchy



Simple abstraction hierarchy



Focusing on a small subset of the hierarchy is less overwhelming.

Sport car rules

- Small
- Two seat
- Luxury
- High speed

Simple abstraction hierarchy

Correct classification

Abstraction and software development

Developing an abstraction of the problem is a necessary first step of all software development.

Object Oriented Approach

Objects & Classes

State / Data / Attributes

Behavior / Operations / Methods

What is OOP?

The logical approach used in software engineering that describes how a programming language is implemented.

It is a **programming paradigm** where developers think of a program as a **collection of interacting objects**

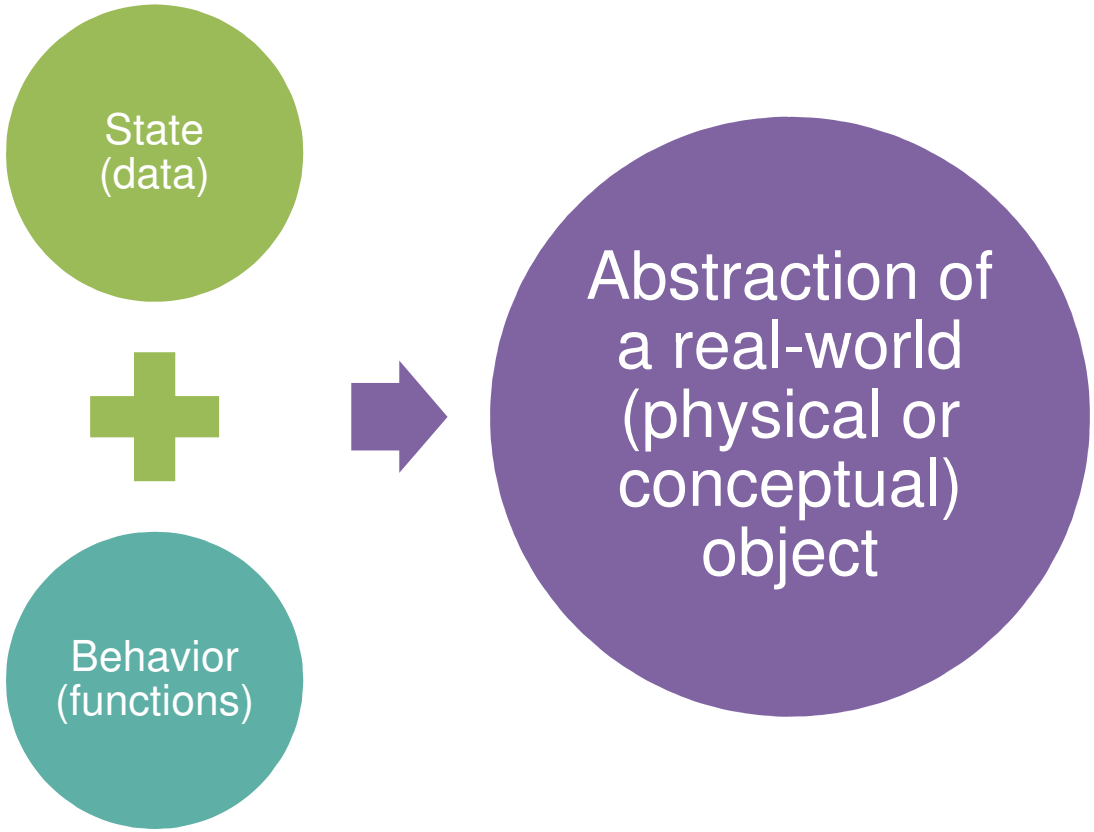
What is a object?

(1) something material that may be perceived by the senses; (2) something mental or physical toward which thought, feeling, or action is directed.

Merriam-Webster's Collegiate Dictionary

Physical objects	Conceptual objects
• Person • Student • Professor	• Class • Grade • Age

What is a software object?



State / Data / Attributes

State is a group of object's attribute values

Attributes are data elements used to describe an object

State / Data / Attributes

Student attributes

- Name
- Birth date
- ID
- Program

Student state

- **Name**: Smith Garden
- **Birth date**: 22/JUL/1970
- **ID**: 649851
- **Program**: Computer Science

Behavior / Operations / Methods

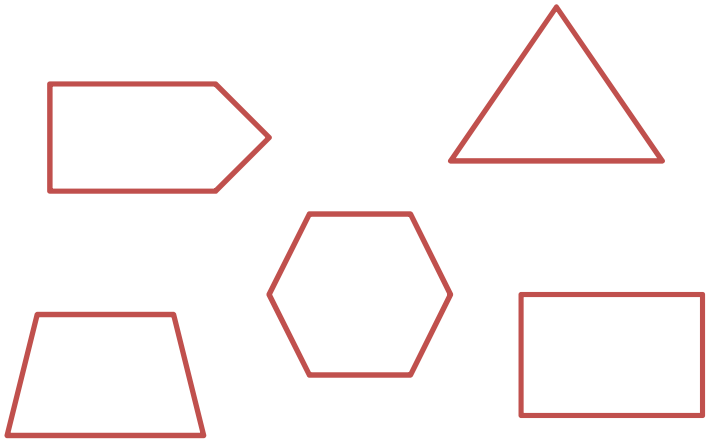
Operations are	
the things that an object does to modify its attribute values	things that an object does to access its attribute

What is a class?

Is a list of **common attributes and behaviors** for a set of similar objects.

```
Class Shape
      Area
      Perimeter
      Angles
      Edges

      Get Area
      Get Perimeter
      Get List of Angles
      Get List of Edges
```

Class example



Objects examples

Objects in Java

Creating classes
Instantiation
Encapsulation

Creating the Student Class

Attribute	Type
id	integer
name	String
surName	String
birthDate	Date
papa	double
advisor	???
courses	???

```
public class Student {

    int id;
    String name;
    String surName;
    Date birthDate;
    double papa;
    // advisor ???
    // courses ???

    // Method declarations goes here
}
```

A class definition is like a class construction **template**

Instantiation

Is the process by which an object is **created in memory** based upon a class definition.

Attribute	Type	Value
id	integer	To be determined
name	String	To be determined
surName	String	To be determined
birthDate	Date	To be determined
papa	double	To be determined
advisor	???	To be determined
courses	???	To be determined

Class definition

Instantiation

```
public class StudentTest {

    public static void main(String[] args) {

        Student myStudent = new Student();

        myStudent.name = "Bruce Wayne";

        myStudent.talk();
    }
}
```

Instantiation

```
public class Student {

    int id;
    String name;
    String surName;
    Date birthDate;
    double papa;
    // advisor ???
    // courses ???

    void talk() {
        System.out.println("My name is: " + this.name);
    }
}
```

Encapsulation

Is one of the four **fundamental principles** of object-oriented programming.

Is a process of **hiding all the internal details of an object** from the outside world

Is a **protective barrier** that prevents the code and data being randomly accessed by other code or by outside the class

Encapsulation

```
public class Student {

    private int id;
    private String name;
    private String surName;
    private Date birthDate;
    private double papa;
    // advisor ???
    // courses ???
```

name has private access in lesson.Student
--
(Alt-Enter shows hints)

```
myStudent.name = "Bruce Wayne";

myStudent.talk();
```

Encapsulation - Accessor and mutators

```java
public class Student {

    private int id;
    private String name;
    private String surName;
    private Date birthDate;
    private double papa;
    // advisor ???
    // courses ???

    public String getName() {
        return "My name is: " + this.name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Accessor

Mutator

Using Accessor and mutators

```
public class StudentTest {

    public static void main(String[] args) {

        Student myStudent = new Student();

        myStudent.setName("Bruce Wayne");

        System.out.println(myStudent.getName());
    }
}
```

Encapsulation benefits

```
public String getName() {
    return "My name is: " + this.name.toUpperCase();
}

public void setName(String name) {

    if (name == null) {
        System.out.println("Invalid name, using default name");
        this.name = "NEW USER";
    } else {
        this.name = name;
    }

}
```

hiding all the internal details

protective barrier

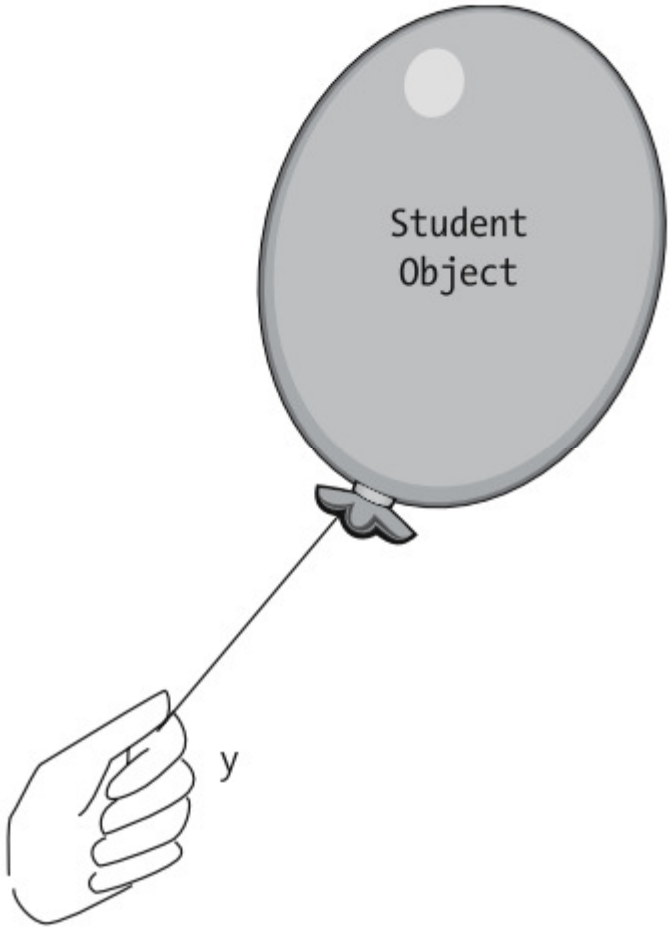
Instantiating Objects: A Closer Look

Working with reference variables

Garbage collector

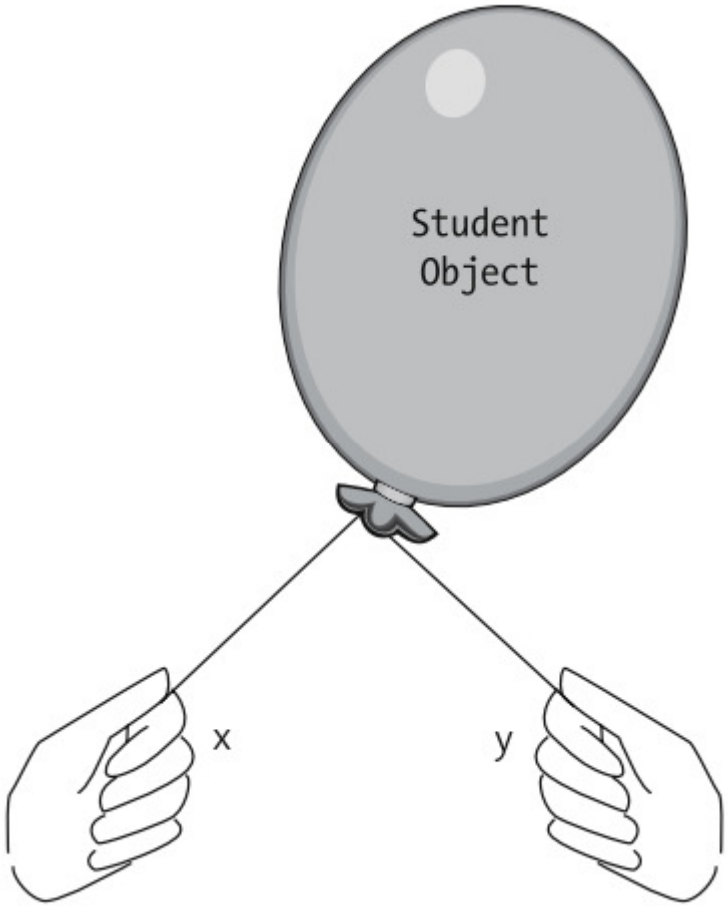
Instantiation - Working with reference variables

```
Student y = new Student();
```



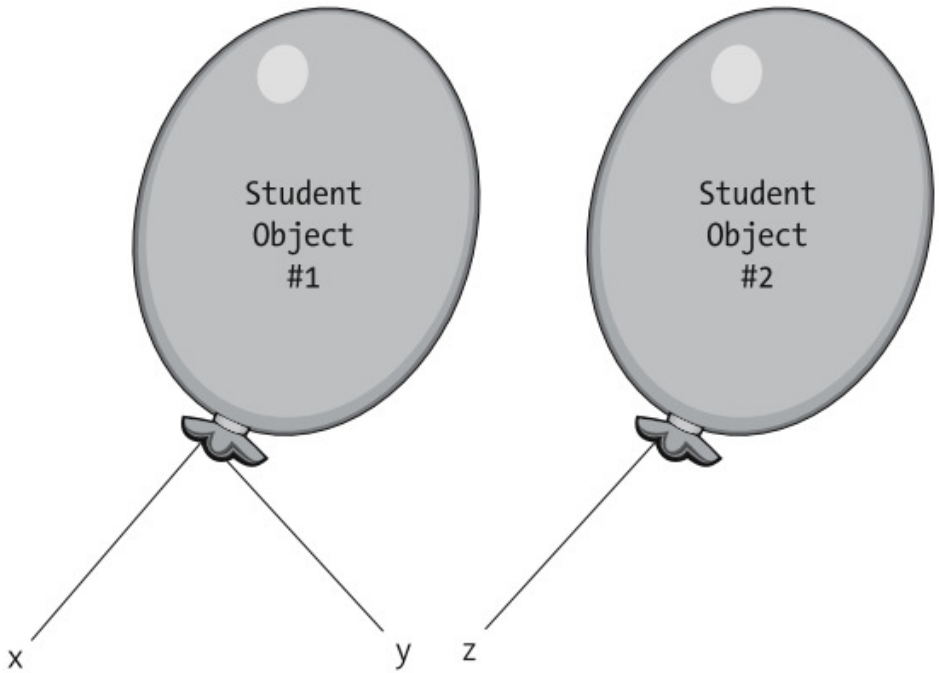
Instantiation - Working with reference variables

```
Student y = new Student();

Student x;
x = y;
```



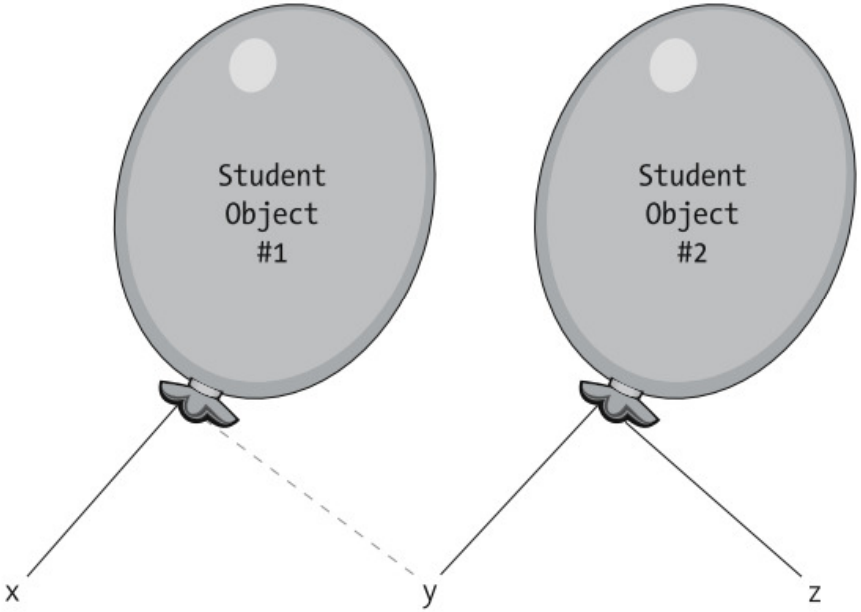
Instantiation - Working with reference variables

```
Student y = new Student();

Student x;
x = y;

Student z = new Student();
```



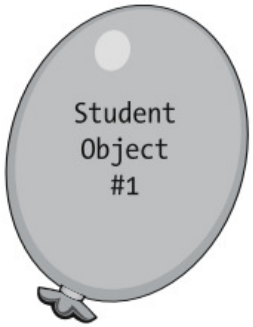
Instantiation - Working with reference variables

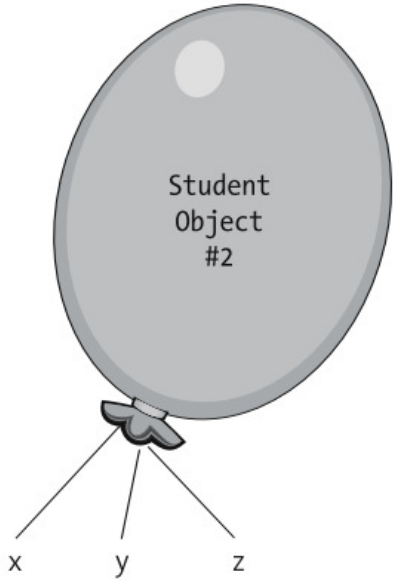
```
Student y = new Student();

Student x;
x = y;

Student z = new Student();

y = z;
```



Instantiation - Working with reference variables

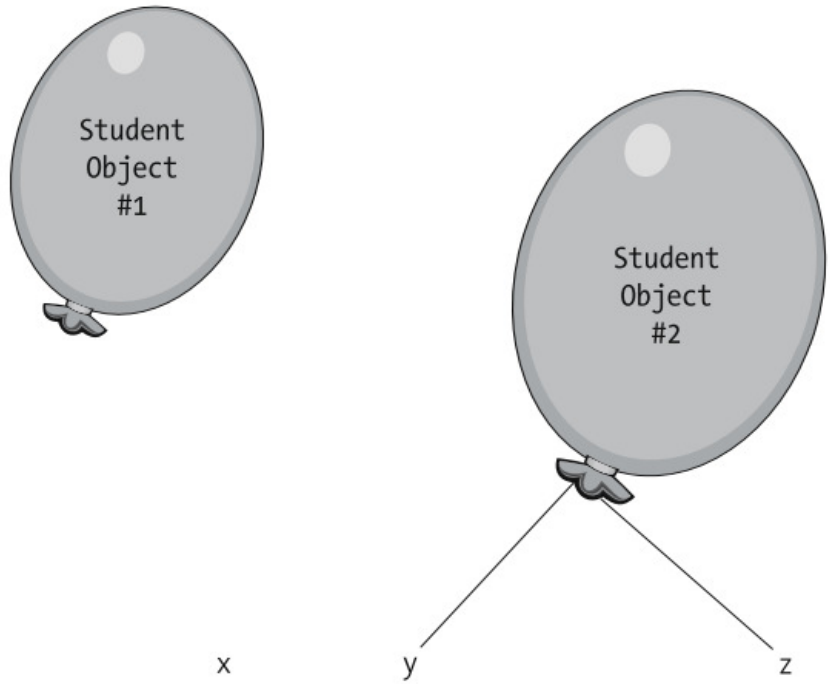
```
Student y = new Student();

Student x;
x = y;

Student z = new Student();

y = z;

x = z;
```





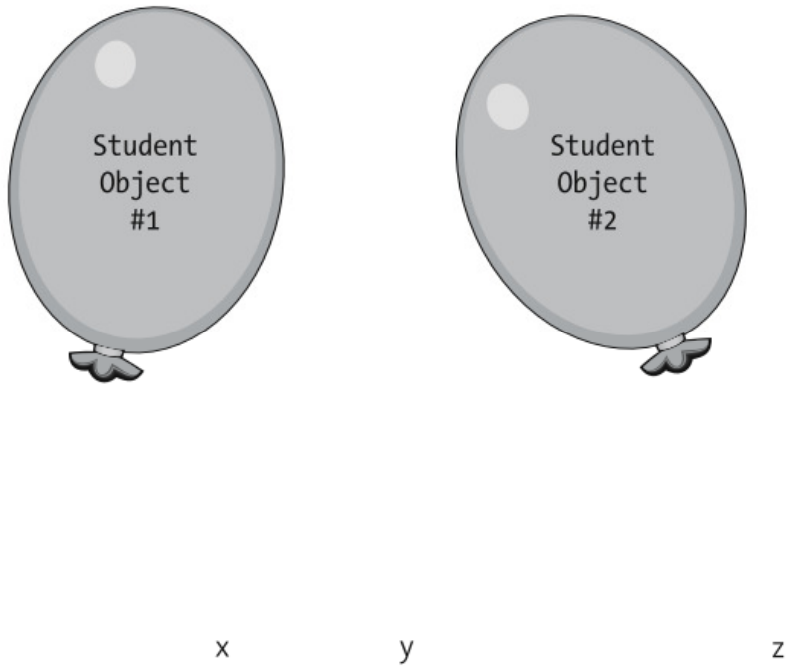
Instantiation - Working with reference variables

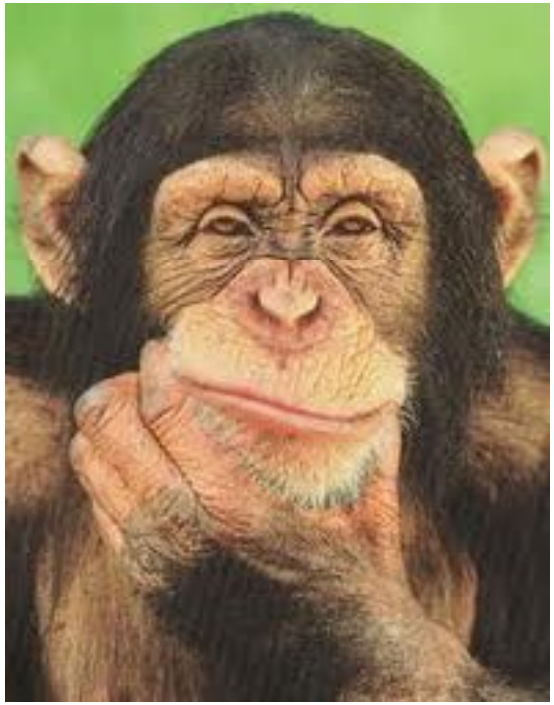
```
Student y = new Student();

Student x;
x = y;

Student z = new Student();

y = z;

x = z;

x = null;
```



Instantiation - Working with reference variables

```
Student y = new Student();

Student x;
x = y;

Student z = new Student();

y = z;

x = z;

x = null;
y = null;
z = null;
```





Where do my balloons will go?

Garbage collection

Garbage collection

- If there are no remaining **active references to an object**, it becomes a candidate for garbage collection.

- Garbage collection occurs whenever the JVM determines that the application is **getting low on free memory, or when the JVM is otherwise idle**.

Class activity

1. **Abstract the model to submit the grades of a student in the SIA (Classes, behaviors, attributes, etc)**
2. Create a Java project in NetBeans or Eclipse
3. Create the Java classes of the proposed model
4. Encapsulate the classes

References

- [Barker] J. Barker, *Beginning Java Objects: From Concepts To Code*, Second Edition, Apress, 2005.
- [Deitel] H.M. Deitel and P.J. Deitel, *Java How to Program*, Prentice Hall, 2007 - 7th ed.
- [Sierra] K. Sierra and B. Bates, *Head First Java*, 2nd Edition, O'Reilly Media, 2005.