

Computer Science II

CS 1101 – Introduction to Programming

Object oriented programming (OOP)

- OOP is a *paradigm* or *philosophy* for how programming should work
- Contrast with the more traditional *procedural programming*:
 - Relies on procedures (or “methods”) to write modular/re-usable code
 - Data (variables) and functions (methods) are separate
- Meanwhile in *object oriented programming*:
 - Relies on objects to write modular/re-usable code
 - Data and function are combined when they are related

Objects

- Objects are a combination of related *attributes* (data) and *operations* (functions)

Attributes define the object's characteristics and properties

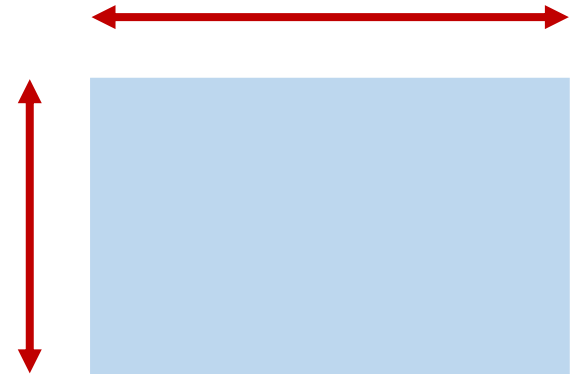
Operations define what can be done with the object and how it changes

Objects – attributes and operations

For example: what are the **attributes** and **operations** for a Rectangle object in a graphical program?

Attributes:

Operations:



Objects – attributes and operations

- Attributes of an object should be chosen based on the situation in which they will be used
- Beware of stale attributes!
 - Some attributes are more fundamental than others
 - This again depends on how you will use your object

Objects – attributes and operations

- In Java, Objects are defined with *class* files
 - These are the blueprints for creating Objects
- Each class file contains details about variables (object *attributes*) and methods (object *operations*)

Objects – attributes and operations

- Class files are not objects: they *define* objects that are then created elsewhere
- Writing a class file is a bit like defining your own custom variable type
 - Instead of only storing an `int` or a `double`, you can store as much data as you want, of any type(s)
 - You can also write relevant methods for operating on this data

Objects – thinking with objects

Object	Attributes	Operations
Circle ..in a graphical program		
Bank account ..in a financial program		
Employee .. in a HR payroll program		

Objects – thinking with objects

Object	Attributes	Operations
Student ..in a university's enrollment program		
Clock ..in a cell phone app		
Enemy Monster ..in a computer game		

Objects – thinking with objects

- When designing objects, some attributes and operations may need to be internal-only
- Others should be available for anyone to see and use
- This depends on both the object and the application (i.e., how the object will be used)

Objects in Java

There are three steps for working with Objects in Java

1. **Design** the Object
2. **Create** the Object
3. **Use** the Object

Step (1) is what we do within a class file

Objects in Java - designing

- To create an Object, we first need to have a clear description:

Create a class file that defines a rectangle object that can change its own length and width, and can compute its own area.

- Start in Java with a single line that declares and names the class:

```
public class Rectangle {
```

Objects in Java - designing

```
public class Rectangle {  
    private double width;  
    private double length;  
  
    public Rectangle() {  
    }  
  
    public void setWidth(double w) {  
        width = w;  
    }  
  
    public void setLength(double l) {  
        length = l;  
    }  
}
```

Instance variables

Constructor method

Mutator ("set") methods

Objects in Java - designing

- *Instance variables* are the object's attributes
 - These are internal to the object, so they are usually private
- The *constructor method* is a special method used to create new objects
- The *mutator methods* are used to modify the object's instance variables
- (Not seen:) *Accessor methods* are used to retrieve the values in the object's instance variables

Objects in Java - designing

```
public class Rectangle {  
    private double width;  
    private double length;  
}
```

- Instance variables *belong* to the object
- They are declared within the class definition, not within a method!
- Used this way, the variables can be referenced by *any method* inside this class!

Objects in Java - designing

```
public class Rectangle {  
    public Rectangle() {  
    }  
}
```

- The constructor method tells Java how an object should be created
- Java provides an invisible *default* constructor method with no parameters, but it is good practice to put one in yourself

Objects in Java - designing

```
public class Rectangle {  
    public void setWidth(double w) {  
        width = w;  
    }  
}
```

- Mutator methods (or *set methods*) allow external classes to modify the values within instance variables
- Without mutator methods, there would be no way to access these variables from outside the object!