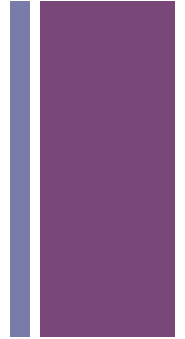


Objects and Classes

+ PrintWriter Example

```
PrintWriter fileOut = null;
try {
    fileOut = new PrintWriter(new FileWriter(outFileName));
} catch (IOException ioe) {
    ioe.printStackTrace();
}
if (fileOut != null) {
    fileOut.print("Point to me... ");
    fileOut.println("<---- but not to me.");
}
if (fileOut != null) {
    fileOut.close();
}
```

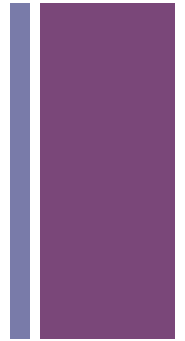
+ Basic try/catch



```
■ try {
    some code that raises exception(s)
} catch (<Name of exception> e) {
    <exception handling code>
    // Some useful commands...
    e.printStackTrace();
    System.exit(1);
}
```

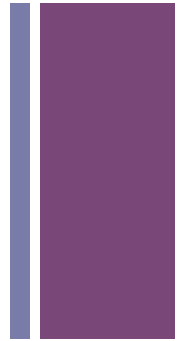
+ Read text from a file

- **Place the data file (e.g. someData.txt) in a Data folder in the Project folder**
- **Import**
 - `java.util.File;`
 - `java.util.Scanner;`
- **Open the file an a Scanner stream:**
 - `Scanner input = new Scanner(new File("Data/someData.txt"));`
- **Scanner methods:**
 - `hasNext():` returns true, if there is line to read, false o/w
 - `nextLine():` returns next line in file as a String
 - `close():` closes the input stream



+ A general code schema to read from a file:

- ```
while (input.hasNext()) {
 String line = input.nextLine();
 <process line>
}
```

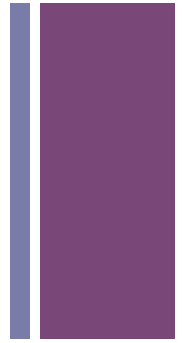


# + Need to handle Checked Exception: (FileNotFoundException)

```
■ try {
 Scanner input = new Scanner(new File("..."));

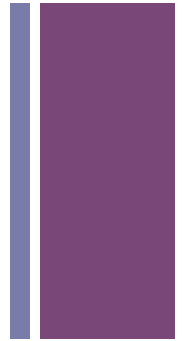
 while (input.hasNext()) {
 String line = input.nextLine();
 <process line>
 }
 input.close();
} catch (FileNotFoundException e) {
 System.out.println("Error in opening data file.");
 // Some useful commands...
 e.printStackTrace();
 System.exit(1);
}
```

# + Object Oriented Programming



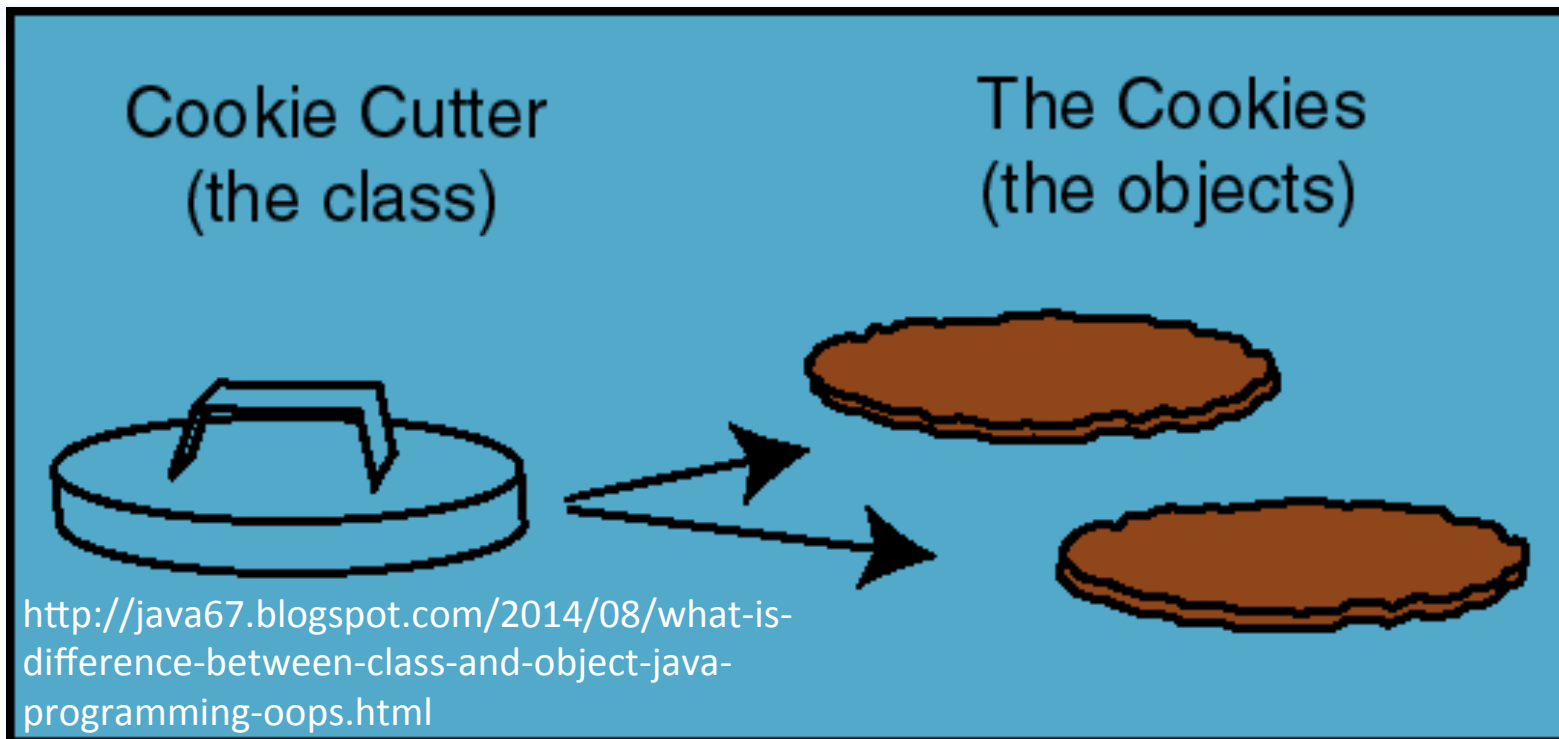
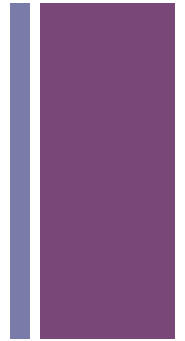
# + What is an Object?

- An **object** is an **instance** of a **class**.
- What is an **instance**?
  - An **instance** is a distinct example of the class that
    - is in memory
    - has specific **assignments** for the **variables declared by the class** it represents.
    - has functionality based on the class.
- What is a **class**?
  - A complex data type.
  - The design for objects of its type.

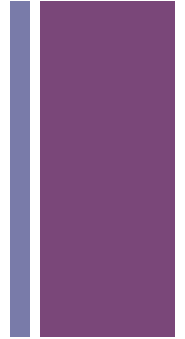




# + Class vs. Object



# + Classes/Objects we've seen

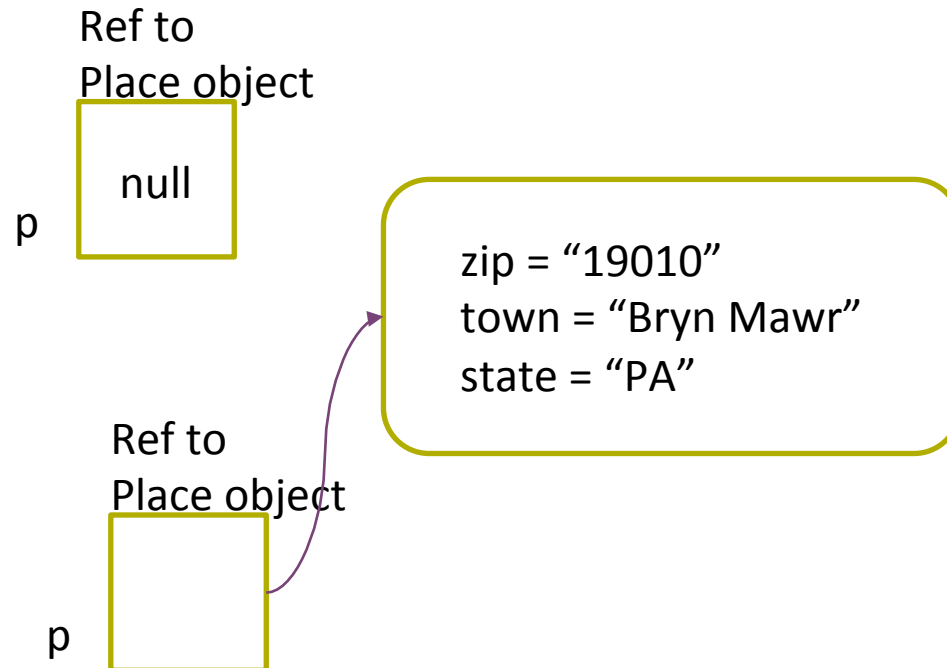


- **System – (Class)**
  - System.out (reference to Object)
  - System.in (reference to Object)
- **String (Class)**
  - args[0] (reference to an Object)
  - args[1] (reference to an Object)
- **Scanner (Class)**
  - Scanner scan; // declaration of reference to Object
  - scan = null; // initialization of Object reference to null
  - scan = new Scanner(System.in); // Instantiation of object

# + Example Class: Place

```
■ Place p;
 p = new Place("19010",
 "Bryn Mawr",
 "PA");
```

```
String z = p.getZip();
String t = p.getTown();
String s = p.getState();
```



| Place                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attributes</b> <ul style="list-style-type: none"><li>- zip</li><li>- town</li><li>- state</li></ul>                                                                                                                                                                                                                            |
| <b>Methods</b><br><b>Constructors</b> <ul style="list-style-type: none"><li>+ Place(String, String, String)</li></ul><br><b>Accessors</b> <ul style="list-style-type: none"><li>+ getZip()</li><li>+ getTown()</li><li>+ getState()</li></ul><br><b>Print Method</b> <ul style="list-style-type: none"><li>+ toString()</li></ul> |