OBJECTS AND CLASSES

Overview

- What is an Object?
- What is a Class?
- Characteristics of Objects
- Static Methods
- Constructors
- Cookie Cutters
- Dot Notation

What makes an object

- Made of tangible material (plastic, metal)
- An object holds together as a single whole (the whole pen).
- An object has properties (the color of the pen, where it is, how thick it writes...).
- An object can do things and can have things done to it.

An object has

- **identity** (each object is a distinctly different)
- A **state** (it has various properties, which might change).
- **behavior** (it can do things and can have things done to it).

Software objects

- Software objects have **identity**. Each occupies a designated chunk of memory.
- Software objects have **state.** Objects have variables which contain values. These values are determine the state of the object.
- Software objects have **behavior.** They contain *methods* that enable the object to "do things". The object performs- does something when one of its method runs.

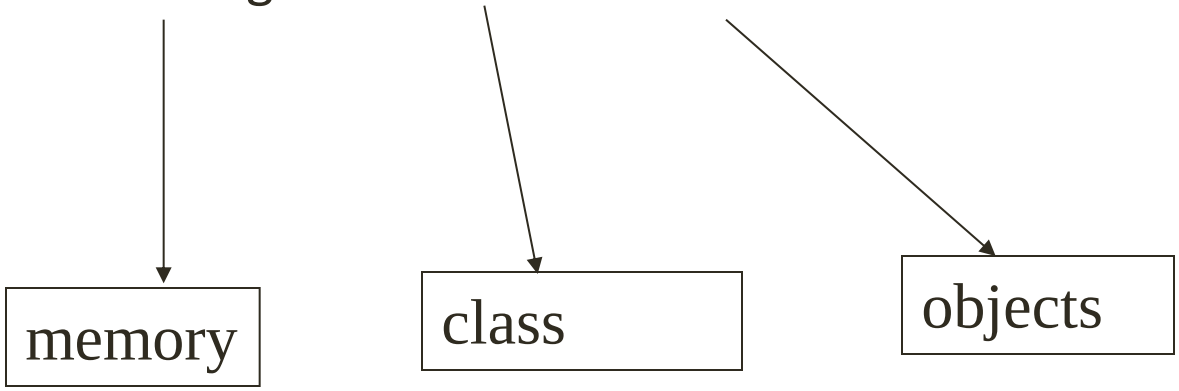
Java classes

- A **class** is a description of a kind of object. A programmer may define a class using Java, and/or predefined Java library classes
- When a Java application is run, objects are created and their methods are invoked (are run).

Use a class to create objects

- A very popular example of illustrating the process of creating a class which will then create many objects is the example of

  cookie dough- cookie cutter- cookies



Cookie cutter vs. cookies

Cookie cutter vs. Cookies

- A cookie cutter has characteristics that are not shared with cookies i.e. made of metal, sharp edges. There may be many cookies made but only one cookie cutter.
- In Java, a **static** characteristic of a class definition is not shared by its objects. There is only one class definition for a given class, so when a program is running, if something is static then there is only one of it.

Static methods

- The methods that belong to a class definition are called **static methods**. A static method is part of a class definition, but is **not part of the objects** it creates.
- *A program can execute a static method without first creating an object!* All other non-static methods exist only when they are part of an object. So an object must be created before they can be executed.

Constructor

- A class is a description of a possible object. A class description is used when an object is created. The new operator is used with a **constructor** to create an object.
- A *constructor* has the same name as the class. Constructors often are used with values (called parameters) that are to be stored in the data part of the object that is created.
- There are usually several different constructors in a class, each with different parameters. However, all the constructors of a class create the same type of object. If we continue with the cookies example – different constructors will create different cookies- with sprinkles, without, larger or smaller size cookies.

Dot notation

- After an object has been constructed it can be changed by using its own methods (not its constructor). ***However, some objects are designed so that their data cannot be changed after the object has been constructed.*** (Class) String objects are **immutable**. *This means that after construction, they cannot be altered.*
- The variables and methods of an object are called the **members** of that object. The members of an object are accessed using **dot notation**.