OOP — Object-Oriented Programming

- An OOP program models a world of active objects.
- An object may have its own "memory," which may contain other objects.
- An object has a set of methods that can process messages of certain types.

Slides by Larry Baker for IB Computer Science.

Also use a slide from Washington CSE 142
https://courses.cs.washington.edu/courses/cse142/21su/

OOP (cont'd)

- A method can change the object's state, send messages to other objects, and create new objects.
- An object belongs to a particular class, and the functionality of each object is determined by its class.
- A programmer creates an OOP application by defining classes.

The Main OOP Concepts:

- Inheritance: a *subclass* extends a *superclass*
  - Ø The objects of a subclass inherit features of the superclass
  - Ø Can redefine them or add new features.
- Event-driven programs: the program simulates asynchronous handling of events
  - Ø Methods are called automatically in response to events.

OOP Benefits

- Facilitates team development
- Easier to reuse software components and write reusable software
- Easier GUI (Graphical User Interface) and multimedia programming

Chapter 3 : Objects and Classes

- In the OOP model
  - Program maintains a world of interacting objects
  - Different types of objects
    - What they can do
    - How they are created
    - How they interact with other objects

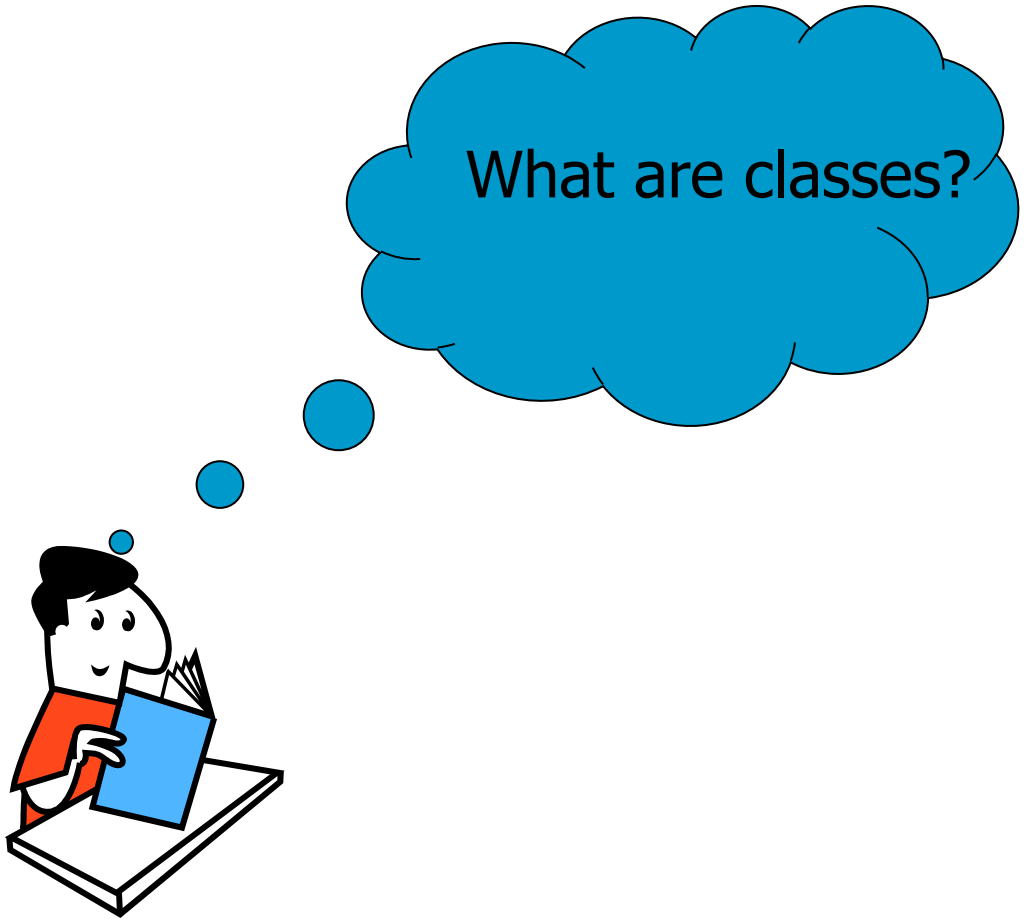
OOP

- An OO program models the application as a world of interacting objects.
- An object can create other objects.
- An object can call another object's (and its own) methods (that is, "send messages").
- An object has *data fields*, which hold values that can change while the program is running.

Objects

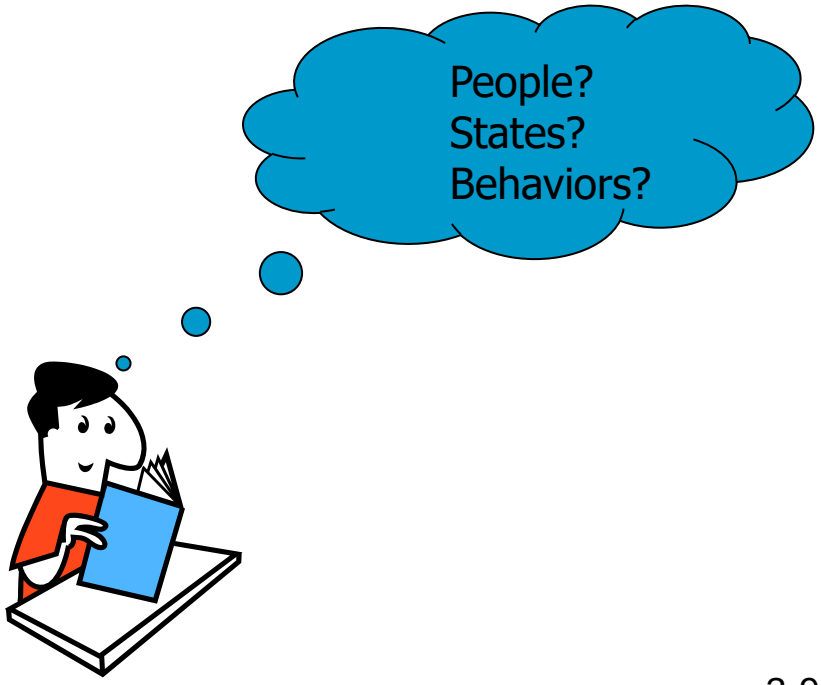
- Can model real-world objects
- Can represent GUI (Graphical User Interface) components
- Can represent software entities (events, files, images, etc.)
- Can represent abstract concepts (for example, rules of a game, a particular type of dance, etc.)

Let's look at the big picture…..



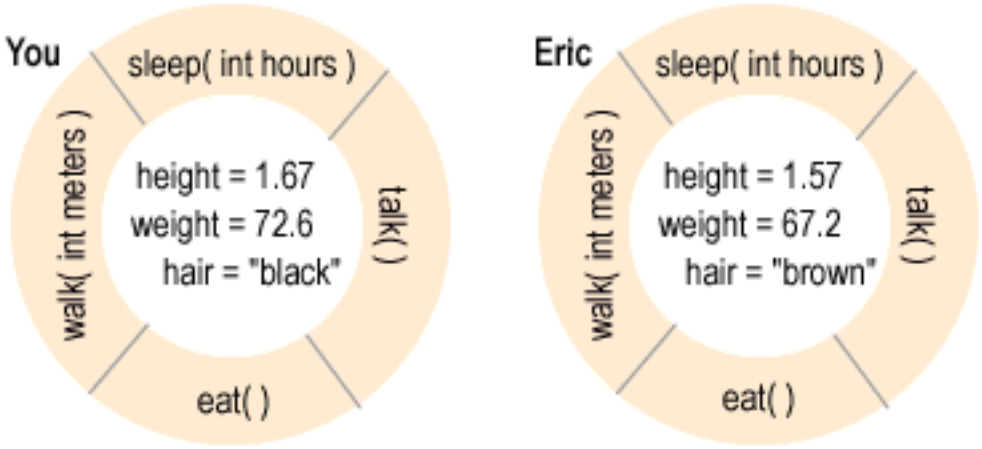
Object-oriented Programming Concepts

- In object-oriented programming, a software object also has state and/or behavior.
- Such an object maintains its state in one or more variables, and implements its behavior with methods.



Object-oriented Programming Concepts

The following picture represents two software objects, one called **You** and the other called **Eric**:

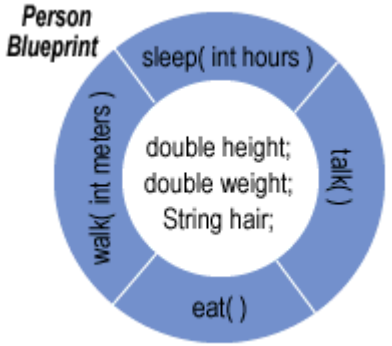


Object-oriented Programming Concepts

- Then Jenny decides she would like to be included in your program, so you again make three variables and four new methods to represent an object called **Jenny**.
- Then the Patel twins want the same, as does Robyn and her two brothers, followed by the Fergusons (three brothers, one sister, fourteen cousins, and an aunt), and before you know it you have hundreds of variables and methods.
- There must be a better way!

Object-oriented Programming Concepts

- There *is* a better way!
- In fact, this is where the power of object-oriented programming comes in.
- Rather than creating new variables and methods for each new person, we instead create a kind of blueprint that tells Java how to make person-objects:

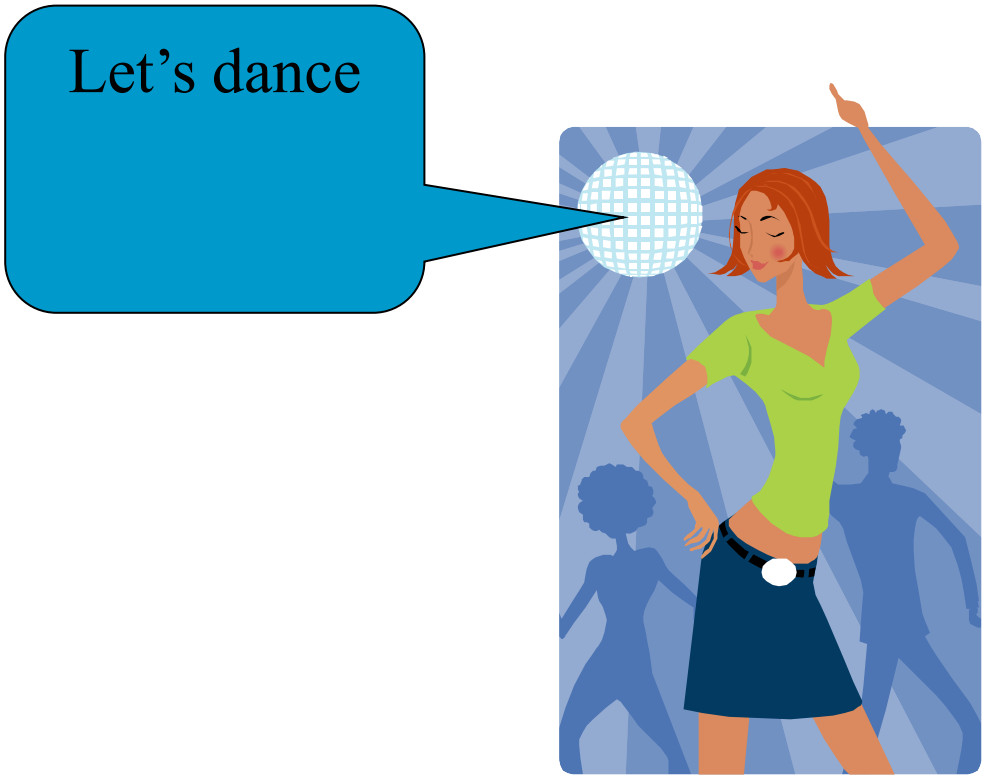


Object-oriented Programming Concepts

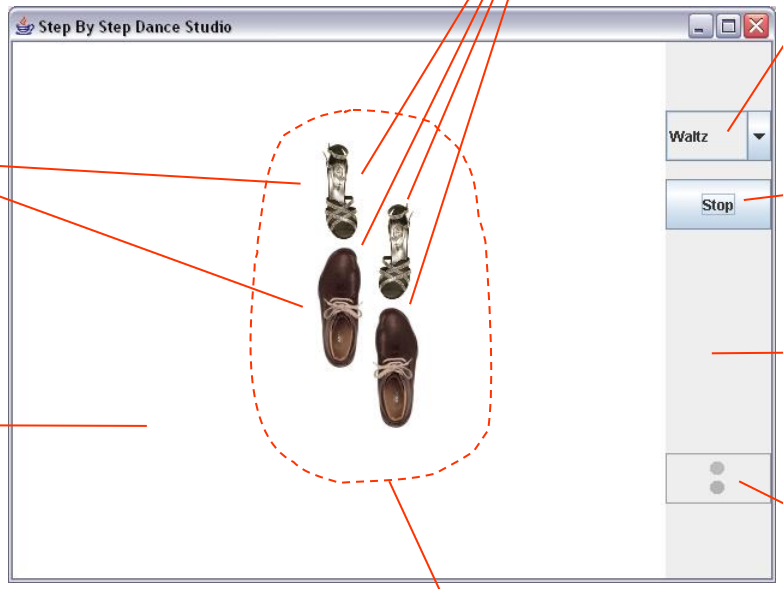
- That being the case, whenever we need a new person-object, we simply ask Java to produce one based on the blueprint — that is, the class — that we have provided.
- We refer to such an object produced by Java either as an *instance* of the Person class or as a Person object (the two phrases are used interchangeably)

Chapter 3 : Objects and Classes

2. Objects in a Program: An example



Objects in the *Dance Studio* Program



Dance Studio window

Band

Foot

Dance selection pulldown list

Dancer

Go / Stop button

Control panel

Dance floor

Positioning button

Waltz, etc.

Dance step

Dance group

The Object Concept

- **procedural programming:** Programs that perform their behavior as a series of steps to be carried out

- **object-oriented programming (OOP)**: Programs that perform their behavior as interactions between objects
  - Takes practice to understand the object concept