Object-oriented concept

Class and Object

- In old style programming, you had:
  - data, which was completely passive
  - functions, which could manipulate any data

Concept: An object has behaviors

- In old style programming, you had:
  - data, which was completely passive
  - functions, which could manipulate any data

- In object- oriented approach, we have **OBJECT**.
  - An object contains both data and methods that manipulate that data.
    - Think of JAVA
  - An object is *active,* not passive; it *does* things

  - An object is *responsible* for its own data
    - But: it can *expose* that data to other objects

What is an Object?

"An object is simply

***a real-world thing* or *concept*"**

- Kendall Scott

"which has

***behavior, state* and *identity*"**

- Grandy Booch

Find Objects!

"An object is simply a real-world thing or concept"

"which has behavior, state and identity"

Q: We are here at CAMT216 for the lecture of SE232.
- what objects do we have?
- what are their behavior, state, and identity?

Thing or Concept

An object can be *"**a thing** in the system"* or *"the system representation of a **conceptual thing**"*.

*"An object need **NOT** represent a physical thing".* It can be **either physical or conceptual**.

Q: We've found some objects for this class/lecture.
- which objects are physical?
- which ones are conceptual?

State

An object has its state, which is *"**all the data** which it currently **encapsulates**".*

The state is represented by "***attributes (or instance variables or data members)*** *each of which has a* ***value****."*

Q: For some of the objects we found,
- define their states. What attributes do they have?
- does the value of each attribute stay the same?

Behavior

Behavior refers to "***the way an object acts and reacts***, *in terms of its* ***state changes*** *and* ***message-passing***".

An object **interprets the message** it receives and reacts to it based on the **current values of its attributes**.

Q: For the objects we found,
- define their behaviors.
- what messages do they receive and how do they react differently?

Identity

Identity defines **what the object is**, which **does NOT change** even when the values of its attributes change and it may behave different.

"the name of the object is not the same thing as the object"

Q: For the objects we found,
- define their identities.
- what does this really mean?: *"the name of the object is not the same thing as the object"*

Example: Mail Box

http://aceoflifewellbeingblog.com/wp-

Example : Mail Box

- State
  - Empty
  - Full
- Behavior
  - Add mail into its collection
- Identify
  - Home address

What is a Class?

"a class is a construct that is used to ***define a distinct type****. The class is* ***instantiated*** *into instances of itself – referred to as class instances, class objects, instance objects or simply objects. A class defines constituent members that enable its instances to* ***have state and behavior****.[1] Data field members (member variables or instance variables) enable a class instance to* ***maintain state****."*

http://en.wikipedia.org/wiki/Class_(computer_programming)

Class

*"A class is the **blueprint** from which individual objects are created."* – *The Java Tutorials*

But.. **why do we need classes**?

Why not just have objects?

Classes .. Why not Objects?

- **Convenience** *(Write Once!)*

  Objects share common characteristics. With class, we can describe a set of objects with the same properties at once.

- **Communication and Checking Errors**

  Classes specify what values are accepted in given contexts, which helps both the human readers and the compiler.

Class.. Some Facts

1. *"every **object** belongs to a **class**"*
2. *"the class of an object determines its **interface**"*
3. *"**A method** is a specific piece of code which implements the operation"*
4. *"the fact that the object provides the operation is **visible** in the object's interface: the method that implements the capability is **hidden**"*

Class.. Some Facts

5. *"the set of **attributes** which an object has is determined by its class"*
6. *"We call the process of creating a new object belonging to class […] **instantiating**"*
7. *"we call the resulting object an **instance** of class"*
8. *"In languages such as C++ and Java, the class has a direct role in the creation of new objects – it actually does the work. […] a class can often be seen as an **object factory**."*

Benefits of OO Design

"*Classes are intended to be **loosely coupled**, **highly cohesive** modules.*"

Easier, cheaper, and more reliable development and maintenance

"*It is inherently natural to **look at the world in terms of objects***"

What if our system views the world in the same way as users see the world? … impact to requirements process, making changes, and user interaction with the system?