Principles of Object Oriented Programming (CS2012)

Lecture 1

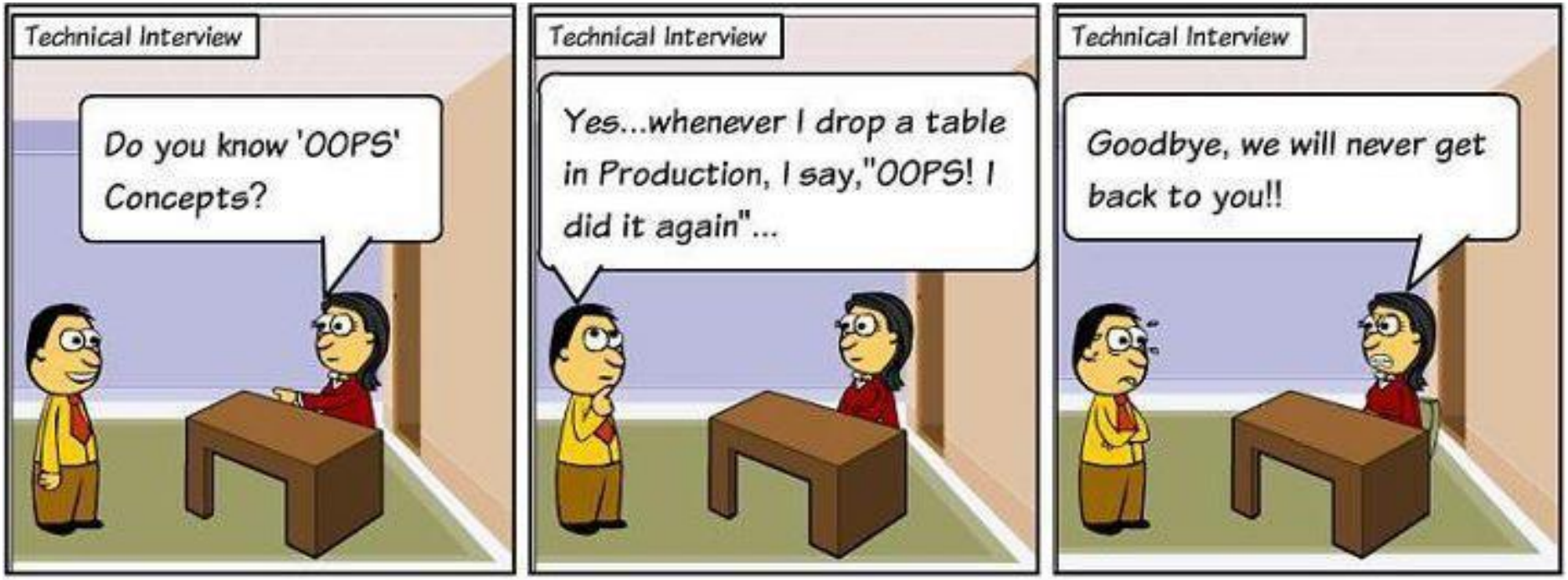
I want you

- NOT to sleep in the class
- Stay away from phone and laptop
- To answer my questions
- To ask me questions
- Not to disturb the lecture

Programming

- Is challenging
- Could be fun
- Could be stressful
- Comes in many forms
- Is an essential knowledge pillar

Object Oriented Programming



Object Oriented Programming

- Objects can be used effectively to represent real-world entities
- For instance, an object might represent a particular employee in a company
- Each employee object handles the processing and data management related to that employee

Object Oriented Programming

- An object has:
  - *state* - descriptive characteristics
  - *behaviors* - what it can do (or what can be done to it)
- The state of a bank account includes its account number and its current balance
- The behaviors associated with a bank account include the ability to make deposits and withdrawals
- Note that the behavior of an object might change its state

Classes

- An object is defined by a *class*
- A class is the blueprint of an object
- Multiple objects can be created from the same class

Objects vs Classes

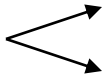
A class
(the concept)

An object
(the realization)

John's Bank Account
Balance: $5,257

Bill's Bank Account
Balance: $1,245,069

Multiple objects
from the same class



Mary's Bank Account
Balance: $16,833

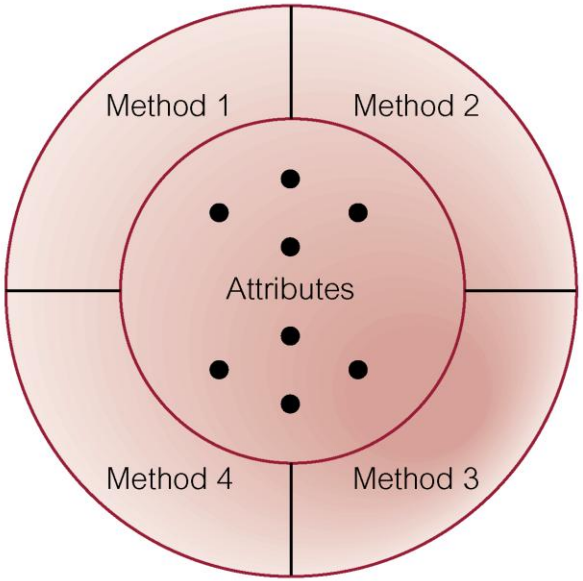
Attributes

- Contain current state of an object
- Attributes can be classified as simple or complex.
- Simple attribute can be a primitive type such as integer, string, etc., which takes on literal values.
- Complex attribute can contain collections and/or references.

Methods and Messages

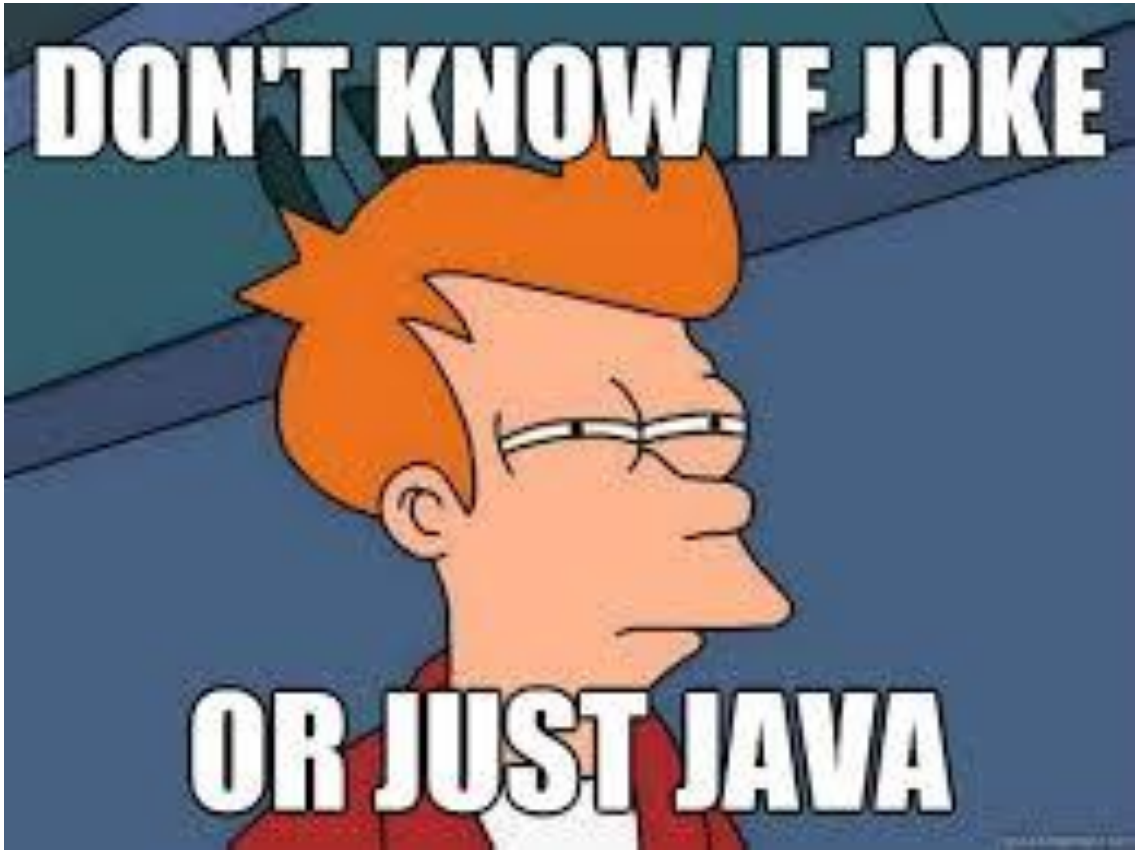
Method: Defines behavior of an object.

Message: Request from one object to another asking second object to execute one of its methods.




```
method void updateSalary(float increment)
{
        salary = salary + increment;
}
```

Java



Basic Java Program

```
public class MyFirstJavaProgram {
    Student aStudent;

    public static void main(String []args) {
        aStudent = new Student(0131234);
        int student_id_number = aStudent. getStudentIdNumber();
    }
 }
public class Student{
    int student_id_number;
    public Student(int student_id_number)
    {
        this. int student_id_number = int student_id_number;
    }
    public int getStudentIdNumber()
    {
        return int student_id_number;
    }
}
```