CMP2103:

Object Oriented Programming (OOP)

Overview

A little housekeeping

- **Office No:**

  Room 3014, New CEDAT Building

- **Class email:**

  OBJECT.PROGRAMMING.CEDAT@GMAIL.COM

- **Office hours:**

  Wednesdays, Thursdays and Fridays, 2:00pm – 4:00 pm

Class Literature

- **Check out**

  http://java.sun.com/docs/books/tutorial/

- **Office hours:**

  Wednesdays, Thursdays and Fridays, 2:00pm – 4:00 pm

Homework and Tests – 40%

- **Homework: 25%**
  - Combination of individual and group work
  - Each group is a maximum of 3 members from the same study program.

  - **Naming homework files:**

    Use protected file formats such as PDF with your file labled in the form:

    CourseCode-YourLastName-YourFirstName-HW#

    Course codes: BIO, CMP, ELE, TEL

    E.g. If Carol OVON is an ELE submitting homework1, your file name will be: ELE-Ovon-Carol-HW1

  - **Late submissions - Homework:**
    - All homework submissions after the deadline are to incur a penalty – 20% off for everyday that the homework is late.

- **Tests: 15%**

  2 tests

Computer Programming

- Algorithm vs. Computer program
  - ✓Algorithm = step-by-step process
  - ✓Computer program = step-by-step set of instructions for a computer.
  - ✓Every computer program is an algorithm

Programming Languages

- Purpose:

Allow programmers to develop software

- Three (3) major families of programming languages:
  - ✓Machine languages
  - ✓Assembly languages
  - ✓High-level languages

Programming Languages cont'd

- Machine languages (ML)
  - ✓ The "native" language of computers.
  - ✓ Comprised of 1s and 0s. Any mistake in ordering the 1s and 0s can cause the program to fail.
- Assembly languages (AL)
  - ✓ Comprised of a set of elemental commands tied to a specific processor.
  - ✓ Assembly language code must be translated to machine language for processing by a computer.
- High-level languages (HL)
  - ✓ Big step forward in easier programming
  - ✓ Syntax similar to the English language.
  - ✓ Subdivided into two major groups – Procedural and Object oriented languages.

Procedural Languages

- Typically refers to early HL languages.

- Such languages focused on structure and were characterized by sequential sets of linear commands.

- Examples:
  - ✓ HTML, C, etc.

Object Oriented Languages

"The real world can be accurately described as a collection of objects that interact".

- Focuses on modeling data as opposed to structure in procedural languages.

- Programmers code using "blueprints" of data models called classes.
- Examples:
  - ✓ Java, C++, Python, etc

Simula

- Considered the first OOP language.
- Simula is the name for 2 programming languages, Simula1 and Simula67 developed in the 1960's.
- Simula was designed for doing simulations and the needs of that domain provided the framework for many of the features of OOP languages today.

Object Oriented Programming (OOP)

- Mainly a programming design philosophy
- Everything in OOP is grouped as self-sustaining "objects".
- This allows for reusability based on 4 main OOP concepts:
  - ✓ Encapsulation
  - ✓ Inheritance
  - ✓ Abstraction
  - ✓ Polymorphism

OOP cont'd

- OOP allows for the definition of the data type of a data structure and the operations/methods (functions) that can be applied to the data structure.
  - This is as opposed to the procedural languages that only define the data type.
  - This way, a data structure becomes an object with both data and functions (methods) in one unit.
  - Also, relationships between objects can be created.
  - E.g.

    Objects can inherit characteristics from other objects.

OOP Basic Terminology

- **Object**
  - Usually a person, place or thing (noun)
- **Method**
  - An action performed by an object (verb)
- **Property/attribute**
  - Characteristics of an object
- **Class**
  - A category of similar objects (such as buildings) , does not hold any values of the object's attributes.

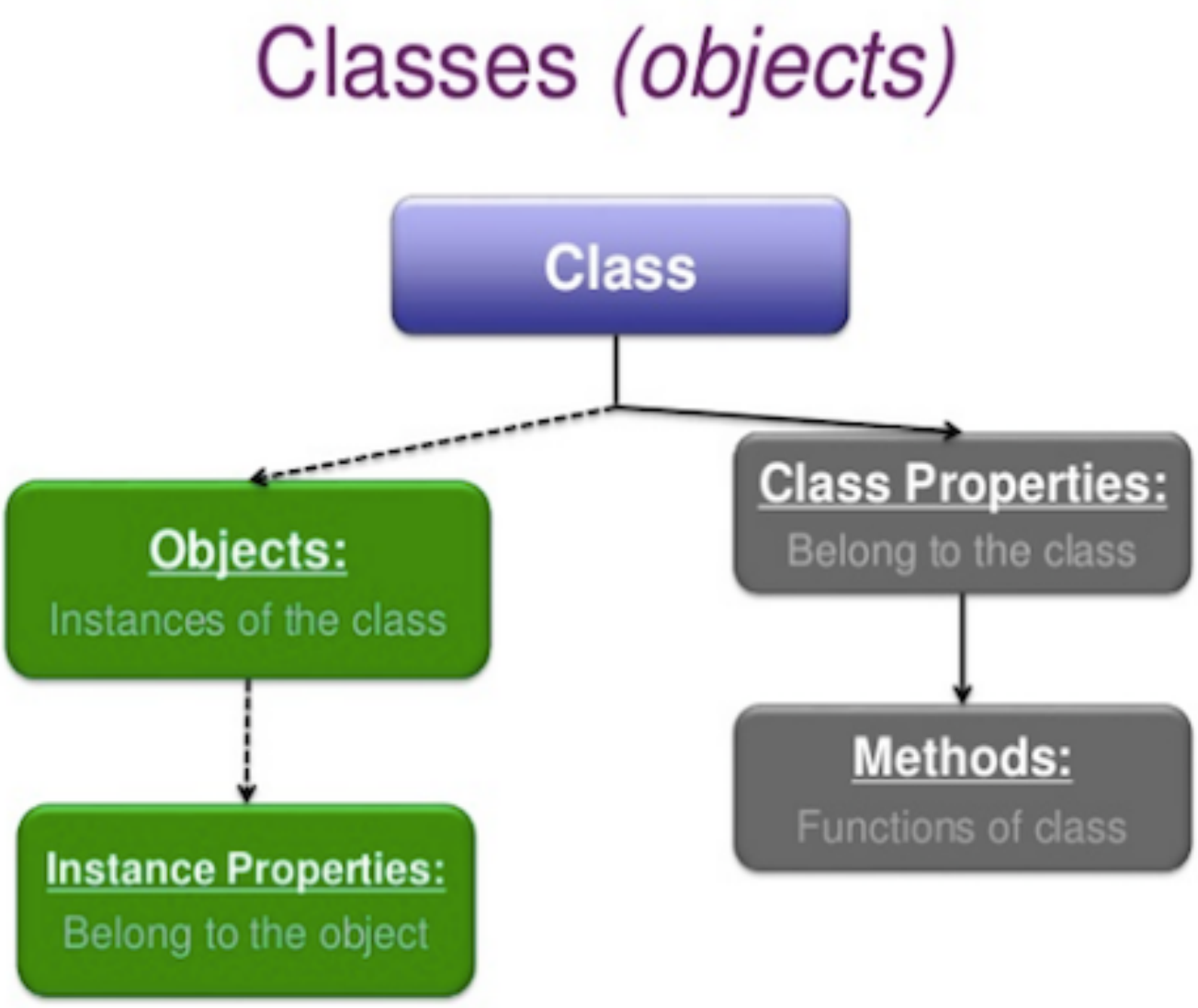
Classes & Objects

- A class is a prototype, idea, and blueprint for creating objects.

- An object is an instance of a class.
  - E.g. in java, classes are defined which in turn helps in the creation of objects.
- A class has a ***constructor*** (special subroutine) for creating objects.

- A class is comprised of 3 things:
  - Name, attributes/properties and methods.

Classes & Objects - graphically



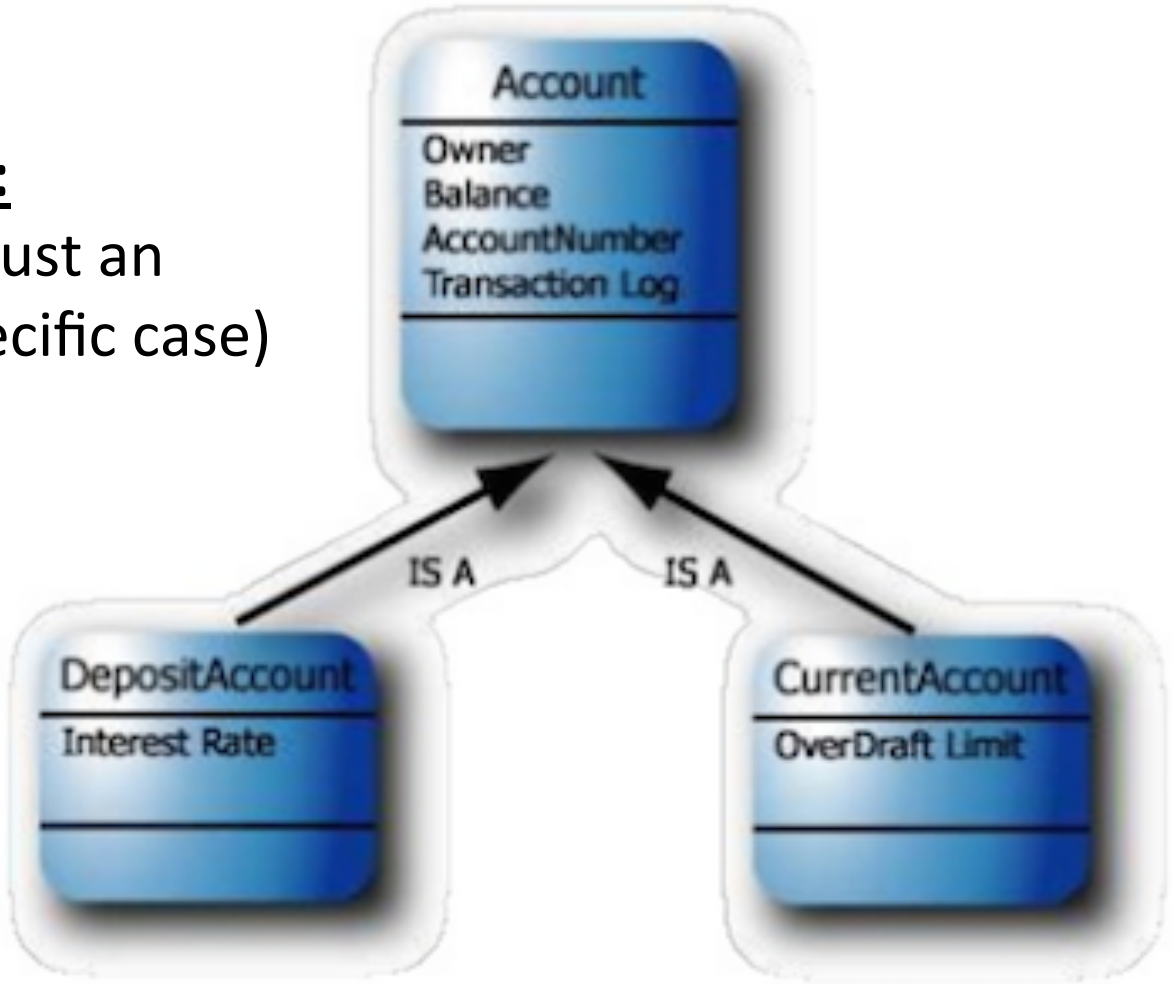
Classes & Objects cont'd

- A class is a definition of objects with the same properties and methods.
- For example:
  - A bank account has . . . . .
    - Account number
    - An owner – account name
    - Balance
    - Transactions
- So . . . . . .
  - How about a current account? a savings account? a foreign currency account?

Bank Accounts - Graphically

Don't forget:

An object is just an instance (specific case) of a class!



Almost everything in the world can be modeled as an object!

Objects - formally

- A formal definition of an object:
  - A computational entity that:
  - ***Encapsulates*** some state
  - Is able to perform functions, or ***methods*** on it's state.
  - Communicates with other objects via ***message passing***.
- A copy of an object from a particular class is referred to as an ***instance*** of the class.
- The act of creating a new instance of a class is referred to as ***instantiation***.
- In short, an object is a class when it comes alive!

Class/Object Technical Differences

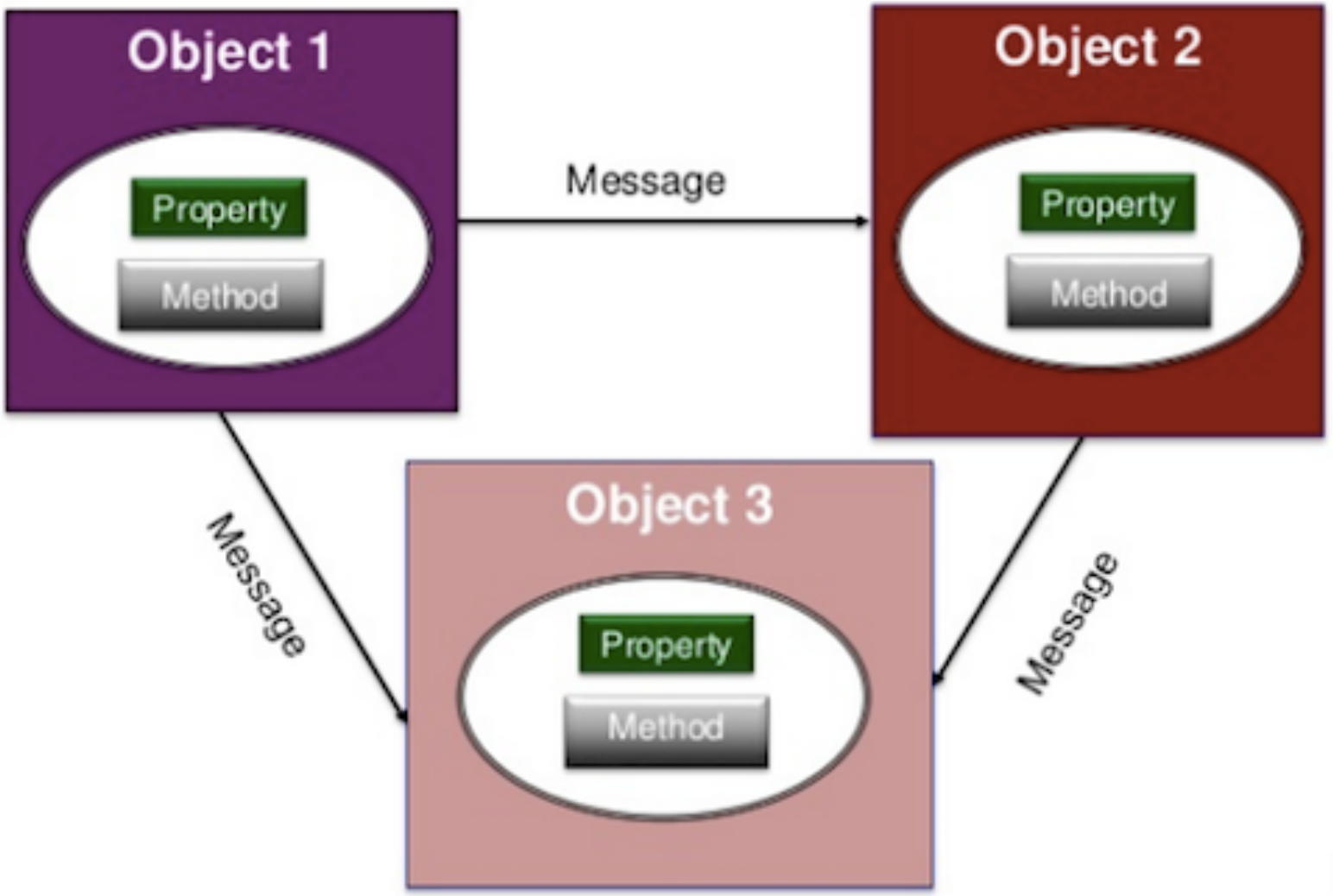
CLASS	OBJECT
Class is a data type	Object is an instance of Class.
It generates OBJECTS	It gives life to CLASS
Does not occupy memory location	It occupies memory location.
It cannot be manipulated because it is not available in memory *(except static class)*	It can be manipulated.

Object is a class in *"runtime"*

Objects Must Collaborate!

- How?
  - Through the exchange of messages
- Why?
  - No one object can carry out every responsibility on its own.
  - Each object is only responsible for its behavior and status.
- Objects are useless unless they can coordinate to solve a problem.

Object Interaction



Homework 1

Due: Thursday, September 20, 2018 by 5pm.

- Define the following as used in programming and give two examples of each:
  - Assemblers,
  - Compilers,
  - interpreters,
- What are the two main differences between compilers and interpreters?

- Develop an algorithm for the operation of Ms. ABC's savings account, current account and USD currency account all held within the same bank. Provide a test case for each of the accounts to show the output of all the associated transactions on the accounts.

- In this course, Java will be our main programming language. For the computing device that you will be using, research and document the installation requirements/procedure and make the installations necessary for your use of Java on your device. In your submission, include screenshots of your installed folder contents and the IDE.