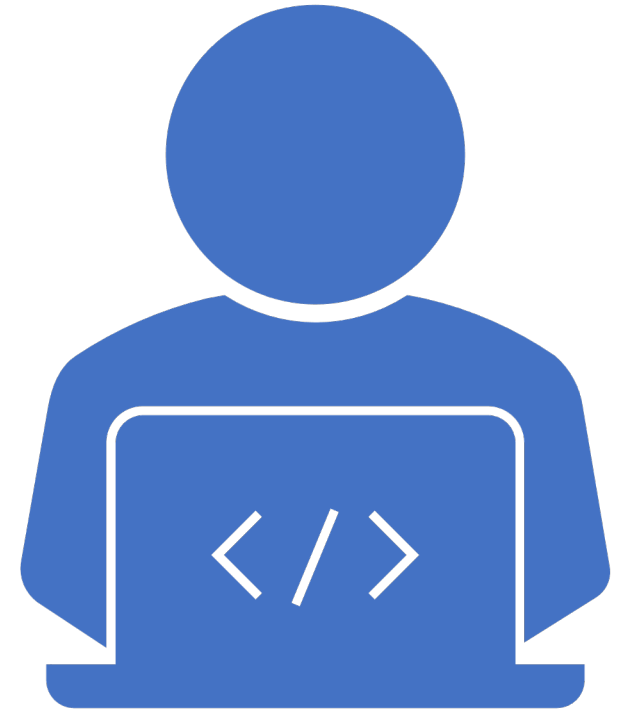


# CMPS 200

## Introduction to Programming Using JAVA

### LECTURE 1 – Introduction

Maurice J. KHABBAZ, Ph.D.



Syllabus already uploaded to MOODLE.  
**! CHECK IT OUT !**



## During this Semester:

### Meeting Days and Times:

- **Section L1:** MWF / 8:00 A.M. – 8:50 A.M.
- **Section L3:** MWF / 10:00 A.M. – 10:50 A.M.
- **Laboratory Sections: B(1, 2, 3, 9, 10, 11) :** Must choose one (AUBsis).

### Meeting Locations:

- Online WebEx Meetings

### Course Instructor for L1 and L3: Maurice J. KHABBAZ, Ph.D.

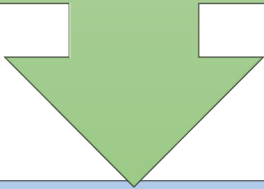
- E-Mail Address: mk321@aub.edu.lb
- Office Hours: Online through pre-scheduled appointments.

# Session Organization & Meetings

AUB - CMPS 200 - M. J. KHABBAZ, Ph.D.

Sunday, August 29, 2021

**Names To Be Announced**  
**will serve as my Teaching Assistants and Lab Instructors**



**During this Semester:**

**Laboratory Delivery Techniques:**

- To Be Announced (TBA) later (may be on-site).
- Will follow the progress of the course.

**Meeting  
Locations:**

- TBA


**Teaching Assistant: TBA**

- E-Mail Address: TBA
- Virtual Office Hours: TBA

Teaching Assistance and Support

### Textbook:

S. Reges and M. Stepp, “Building Java Programs A Back to Basics Approach 5th Ed”, Pearson, 2019.



**Course Type:** Major requirement for all Computer Science students.



**Pre-Requisites:** None.



# Course Textbook, Type and Pre-Requisites

AUB - CMPS 200 - M. J. KHABBAZ,  
Ph.D.

Sunday, August 29, 2021



1. Construct working programs with primitive data types, conditional and iteration.
2. Use a programming environment to edit, compile and test programs.
3. Construct working programs with linear (*e.g.* arrays, lists) and 2D data structures.
4. Understand function signatures and calling conventions to call and use functions and libraries.
5. Write functions / methods that implement given signatures.
6. Decompose a problem into a well-structured set of functional abstractions.
7. Develop strategies for incremental development and debugging of programs.
8. Use classes, including classes from standard libraries, to organize the data that programs operate on and to manage program complexity.
9. Implement classes from well-defined specifications of their interface.
10. Design classes and programs from informal problem specifications.
11. Read small-to-medium-sized programs in order to predict their behavior and modify them.
12. Write programs that adhere to standards of style and documentation.

# Course Learning Outcomes

# Course Topics

## Topics Covered

**Introduction To JAVA:** data types, variables, operators, branching

**Basic Program Building Blocks:** bindings, flow control, iteration, strings, I/O

**Simple Programs:** building and testing

**Functions:** decomposition, abstraction, scope, arguments, specifications, files

**More Building Blocks:** arrays, ArrayList, recursion

**Good Programming Practices:** testing and debugging, exceptions, etc...

**Classes and Inheritance:** OOP, class instances, methods, hierarchies, custom types

**Larger Programs:** structure, documentation, modules

static typing, numerical arrays, more on recursion

algorithms, searching, sorting

# Assessment Measures



Assignments

10%



Labs

5%



Finger Exercises

5%



Exam 1

20%



Exam 2

20%



Final Exam

40%

- No make-ups are allowed.
- Missing an assignment will result in a grade of zero on the missed assignment.
- Missing the Midterm will transfer its percentage to the Final Exam.
- Missing the Final Exam will result in failing the course.

# Some Preliminary Brainstorming

Why are we here?

Why are you taking this course?

Why you shouldn't take this course?

What are characteristics of a great first programming language?

Why is JAVA an excellent first choice of a programming language?





# What is it I owe my students when I teach them to code?

- I think there are 5 key things that students need from a 1st language:
  1. A **great experience on Day 1**.
  2. Ability to (eventually) **program on the Web**.
  3. Ability to **program desktop/mobile applications**.
  4. An eventually **marketable professional skill**.
  5. A **supportive and welcoming community** surrounding the language.

# Which Language Is Best To Start With?

## JAVA? PYTHON?

- Typical 1st program: print something like “Hello World!” on the screen.
- There are many components to successful first experience.

### JAVA does it this way:

```
public class HelloWorld{  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

### PYTHON's much simpler:

```
print("Hello World!")
```

**MUCH SIMPLER**

- **PYTHON** focuses on readability.
- **JAVA** is more explicit and more strict: More powerful in this regard.

# JAVA v.s. PYTHON: More Comparisons

## JAVA

- Compiled Language.
- Faster.
- Real Multithreading:
  - Java Virtual Machine (JVM):
    - Very solid.
- Write Once, Run Anywhere:
  - More portability.
- Has a rich set of libraries:
  - Application Program Interfaces (APIs).
- Nit-Picky and Strongly-Typed:
  - Does not run if errors exist.
- Used in many organizations.

## PYTHON

- Interpretted Language.
- Slower.
- Virtual Multithreading:
  - Global Interpreter Lock (GIL):
    - Not that good.
- May need re-interpretation
  - Less portable.
- Has a rich set of modules:
  - Most written by script kiddies.
- Less picky and Dynamically-Typed:
  - Notifies about errors through warnings and keeps running.
- Not yet that widely employed.

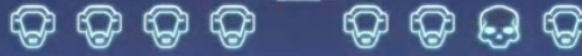
# JAVA: Desktop/Mobile Applications

- PHP, Javascript, ... are limited to web development.
- C is quite primitive for today's sophisticated App developments.
- C++, C# are nice but too verbose and conspicuous.
- PYTHON is at home on local machines.
- **JAVA** is also at home on local machines:
  - Libraries for windowed programs (e.g. [Lightweight JAVA Game Library](#), etc)
  - I used LWJGL to teach teens/graduate students... So much fun!!
- Let me show you a nice game programmed there: [Star Forces: Space Shooter](#)
  - This game is available on the [Google Play Store](#).
  - It also comes pre-installed on some mobile phones (e.g. Huawei, Xiaomi, OPPO, etc).



9904

11:29



Player\_995ac Foxxy\_0xxxxy

Sw00sh  
2.5 km

# JAVA: Desktop/Mobile Applications

AUB - CMPS 200 - M. J. KHABBAZ, Ph.D.

Sunday, August 29, 2021

# JAVA: A Professional Skill

- JAVA used extensively in professional environments
- Language that you can start with
- Want to go low-level?
  - There is a tight integration with hardware
- Want to go high-level?

Great first programming experience  
↓

Dedication/curiosity to evolve professional skills to best fit your goal.

Web Apps?

or PYTHON Web frameworks.

Functional Programming?

with JAVA! There is no need to learn a new language.

# JAVA: Supportive Community

- Personally, I will strive to transform our sessions into welcoming ones!
- I want this class to be the perfect place for you to:
  - Learn the ropes.
  - Meet future colleagues.
- I don't just want to teach you the "HOW?" of things but also the "WHY?"

*"A person who knows how will always find a job.*

*A person who knows why will always be his boss."*

- What I aim at is to trigger your **critical thinking** and your **creativity**.
- If you bare with me, I promise you a fun ride with utmost support.



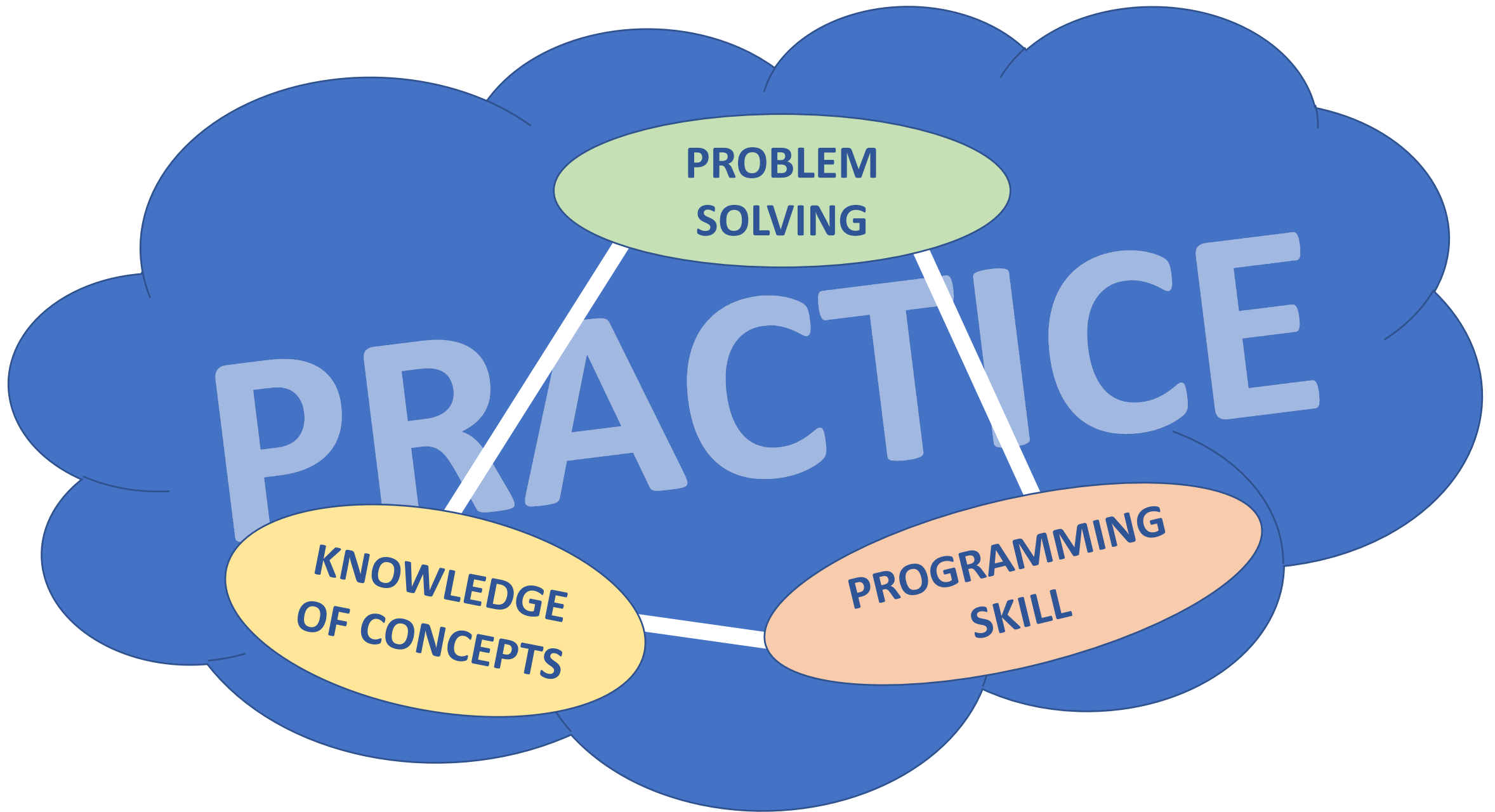
# Together For a Better Learning Experience

- **Traditional way to teach programming:**
  - Give **long assignments** so that students often **pull all-nighters to finish**.
  - This simulates (kind of) what programming is like at a start-up company.
  - Resolving authentic challenges with minimal guidance does mimic some jobs.
  - Causes **drop-off in cognitive performance due to lack of sleep**.
- With **strong instructional guidance**, zero-coding experience students:
  - **Master analytic reasoning and language skills faster and more in-depth**.
- **Studies on Science, Technology, Engineering and Mathematics (STEM):**
  - **Discovery-based learning is not as effective** as strongly guided approaches.
- **I assume zero-coding/programming experience:**
  - I have designed this course to build your skills and **I need your cooperation**.



# Objective Behind Classwork Assignments

- They are very effective at **improving retention and grades.**
- **Attendance** is therefore very important to **keep up with material.**
- **Do not cram to learn all the material a few days before the exam.**
- Work in class and at home to master the material: **!!! PRACTICE !!!**



# Why Hard Questions, Problems, Assignments?

- They are **not hard!!!**
- They might seem **puzzling** ...
- Because you **haven't seen something like that** before ... That's all!!
- This will **trigger** your curiosity and guide you to learn/master stuff.
- **Mental scaffold!**
- Serves to **prime your brain, predispose it** to absorb new information.
- Establishes **fairness** and **equality** among students.

