CSE 11
Accelerated Intro to Programming
Lecture 1

Greg Miranda, Spring 2021

This lecture is being recorded

Fair Notice of Class Recording Announcement

- Each class online lecture for CSE 11, including this one, will be recorded and made available to students asynchronously.

Greg Miranda

- **Started at UCSD in Summer 2018**
    - Taught at private colleges (AICASD, UAT) before UCSD
- **Teaching Background:**
    - CSE 5A/8B/11/12
    - Game Programming, Game Design, Web Programming
- **Background:**
    - CE graduate from UCSD in ’96
    - 20+ years as a Software Engineer
- **When I’m not teaching:**
    - Consulting
    - Taekwondo

Announcements

A01 B01

- Discussion starts Wednesday @ 4pm & 5pm
- Quiz 1 due Wednesday @ 8am
- PA0.5 released today – due Thursday
- PA1 released Wednesday – due 4/7

Coding Experience

- How much coding experience?

17 • A – No coding

22 • B – A little bit of coding

15 • C – Some coding

5 • D – Lots of coding

2 • E

Coding Experience

- Languages – Select All
  - 24 A – Java
  - 8 B – C++
  - 34 C – Python
  - 12 D – JavaScript or other scripting language
  - 12 E – Other compiled language (ex: C)

Topics

- Syllabus
- Canvas
- Questions?

g2miranda @ucsd.edu

email

Experimenting with running Java Programs

→ text file

- How to edit Code?
  - Text Editor → Notepad++, sublime, Visual Studio Code
  - Integrated Development Environment (IDE)
    - eclipse
    - IntelliJ
- How to run Code?
  - Command Line
    - Mac/Linux
    - Windows

compile → javac

run → java

```
class Examples1Lecture {
  int theNumberFive = 2 + 3;
}
```

int → integer
↳ whole #

- Fields → member variable
- Arithmetic Expression
- Output
- Java / Programming Languages
  - Simplest thing you can do with them: calculators
- More Examples

Errors / Error Messages

- Big part of programming
  - Understanding when you made a mistake
  - How to fix the mistake
- Possible mistakes
  - Invalid command
    - Expect that that class is defined in a file with the same name .java
    - Class can't be found – typo in the name or a has mismatch with the name of the class
  - Name of field doesn't match value
    - Pick meaningful names

style → readable code

- Possible mistakes (cont.)
  - Could leave off or forget int
    - Syntax error
  - Error messages do not always match what's wrong
    - Use context of program to figure out what happened
  - Other errors we can get
    - When running programs – practice trying to break them a little bit
      - Remove =
      - Remove ;
      - Remove { or }
- Going to be a lot of times where you make a mistake
  - Typo
  - Copy/paste incorrectly
  - Accidently delete something
  - Or just make a mistake
- Need to practice fixing error messages
  - Use the context of the program to understand the error message
- Errors are a normal part of programming

Arithmetic in Java

```
class Examples1Lecture {
  int x = 2 + 9 * 3;
  int y = 10 / 3;
}
```

- What’s would happen here?
  - What is the value of x?
  - What is the value of y?
  - What about y = 11 / 3; ?
- Using parenthesis

- Order of operations & parenthesizing
  - In many ways Java acts like arithmetic
  - But in other ways, Java does not
    - Division has truncation behavior we do not see in math classes
      - Very common in programming
    - Multiplication & division before addition & subtraction

left → right

↓

stop

New Example

- Create a new file

```
class Examples2 {
  int rate = 20;
}
```

- class – will talk about more later
  - For now: describes a group of fields
- Problem:
  - Calculate the pay you would receive at a certain hourly rate given a number of hours
    - New field: # of hours worked
  - Calculate total pay using Java as a calculator

- Calculate total pay (cont.)
    - Can we use these fields in another calculation?
    - Why is it useful to do this with fields instead of writing this directly?
        - What if:
            - Use same hourly rate, but a number of different weeks to calculate?
        - What if:
            - We want to change the hourly rate?
        - Change once, changes all values
            - Many times, you will have one field whose value can be used in many places
                - Configure how the program works
        - Changing the value in one spot can affect many other places in the program
            - Powerful concept in programming:
                - Define a value in one place
                - Change it by editing the program
                - Watch its changes be reflected in all the other places next time it’s run

- Using this.hourlyRate
  - Call that a field look-up or a field access
    - Looking up the current value of a field that has been defined before