Session 1: Welcome and Chapter 1

Sunday, January 15, 2017 9:04 AM

Syllabus

Session No.	Date	Content	Result
1	Jan. 15	Lecture: Ch1	
2	Jan. 22	Lecture: Ch1(Cont.), Lab1	
3	Feb. 12	Lecture: Ch2	
4	Feb. 19	Lecture: Ch2(Cont.)	
5	Feb. 26	Quiz & Lab2	
6	Mar. 5	Lecture: Ch3	
7	Mar. 12	Lecture: Ch3(Cont.)	
8	Mar.19	Lecture: Ch4	
9	Mar. 26	Lecture: Ch5	
10	Apr. 2	Lecture: Ch6	
11	Apr. 9	Quiz & Lab3	
12	Apr. 15	Lecture: Ch7	
13	Apr. 16	Lecture: Ch8	
14	Apr. 22	Special Prep: Free Response	
15	Apr. 23	Practice Exam	
Final	May. 2	APCS Exam	

Who is Who

- Me -- Jizu Sun
  - 2013 - 2016, Master of Ed. , Nanjing University
  - 2015 - 2016, Internship, Netease, eBay, etc.
  - 2017 - Now, Software Developer, SAP(Shanghai)
- Another Course Lecturer - Weicong Huang
  - 2013 - 2017, Bachelor of Software Eng., SouthEast University,
  - 2017 - 2020, Master of Computer Science, Nanjing University
- And all of you?
  - Why you choose AP Computer Science?
  - Your background
  - Your experiences in computer, esp. programming
- Contacts (Email, WeChat, Phone)

About APCS Exam

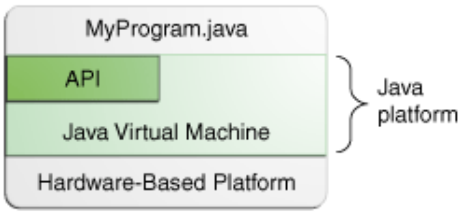
- 2 parts
  - 40 Multiple Choice(MC)
  - 4 Free Response(FR)
- 3 hours
- Paper-based test
- How about Score 5?

Guideline

- It's more than an exam, because it's COMPUTER SCIENCE!
- Spend some time and practice more, after class
- Feedback to me, if you feel uncomfortable
- **Don't trust me, sometimes…**
- **Always listen to your faithful friend, IDE!**

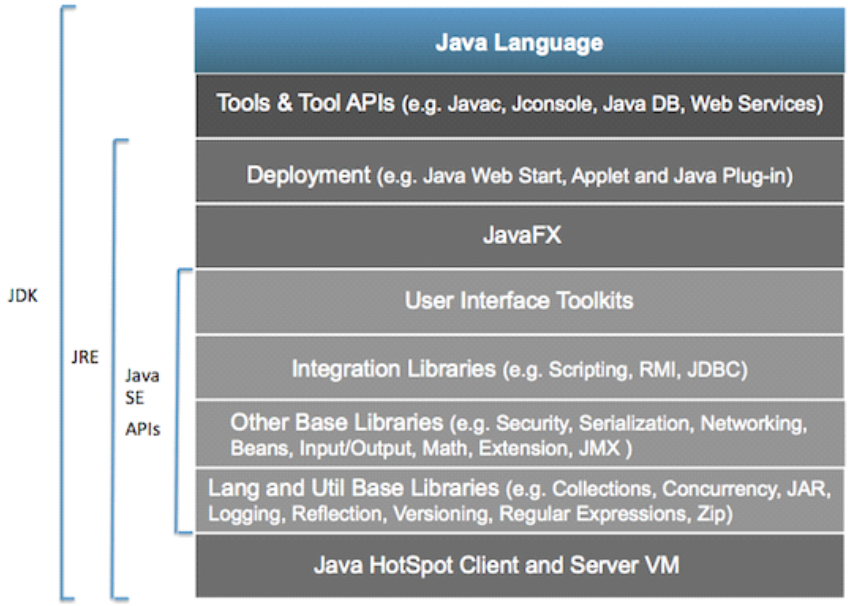
About Java

- History:
  - Designed by: James Gosling
  - Developer: Sun Microsystems (now owned by Oracle Corporation)
  - More on Wikipedia: https://en.wikipedia.org/wiki/Java_(programming_language)
- Getting started:
  - http://www.oracle.com/technetwork/topics/newtojava/overview/index.html
  - http://docs.oracle.com/javase/tutorial/
- Only covers a subset of Java SE:
  - https://apstudent.collegeboard.org/apcourse/ap-computer-science-a (View Java Subset)
- Compare to other languages
  - Need compilation: JavaScript, Python, etc. need not
  - Cross-platform(Once compiled, run everywhere): C, C++, etc. cannot
  - Object-oriented
    - `Class, object`
    - Encapsulation, Inheritance, Polymorphism



Some prep before 2nd class

- BYOD
- Setup Guide:
  https://www.ntu.edu.sg/home/ehchua/programming/howto/EclipseJava_HowTo.html
  - IDE (Integrated Development Environment) : **Eclipse**, or DrJava
  - JDK: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Moodle(Learning Management System): http://edu.nju.edu.cn/moodle/

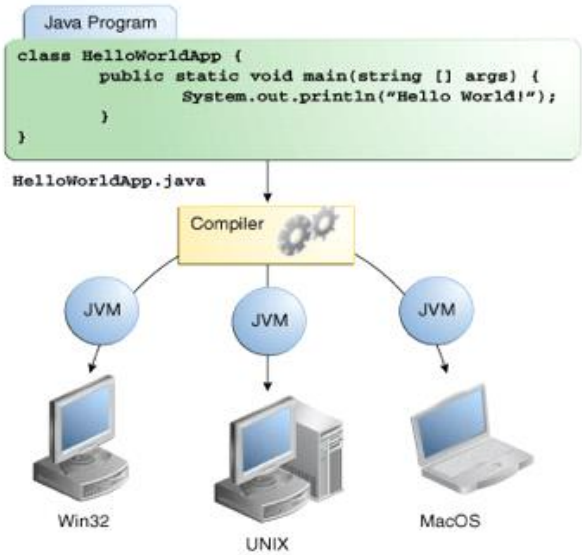


Packages and classes

- *Package: related classes are grouped into packages*
- import is not expected to write in APCS exam
  - `import packagename.subpackagename.ClassName`
  - `import packagename.subpackagename.*`

A simple Java program

- Must have at least one class, which contains the main method
- Demo in **Eclipse**
- Source code ---(compiler)---> bytecode ---(JVM)---> RUN ON MACHINE
- Comments
  - `/* here */` and `// there`



Types and identifiers

- Identifiers: a name for a variable, constant, user-defined method, or user-defined class
  - Letters, digits, and the underscore
  - May not begin with a digit
  - Case-sensitive

    - Concise and self-documenting
    - `Lowercase or uppercase`
        - *`anotherVariable, anotherMethod, AnotherClass`*
    - Reserved words
- Built-in types / primitive types
    - **`int, double, boolean`**
    - `Numerical casts: int <-> double`
        - `int -> double`
            - `implicit cast: automatically cast`
        - `double -> int`
            - `without a cast cause a compile-time error`
            - `With an explicit will cause truncation: (int) cost`
- ***Variable***: is a named memory location can be assign a value. Further, the value of a variable can be changed during the execution of a program
- Declare a variable, and initialize it, or assign later **(declaration, initialization, assignment)**

Storage of numbers

- Integers: int (32bit)
    - The sign bit(Most significant bit, MSB): 0 for non-negative, 1 for negative
    - Two's complement
    - n-bit integer: can store signed integers from -2^{n-1} to $2^{n-1} - 1$
    - `Integer.MIN_VALUE, Integer.MAX_VALUE`
- Floating-point numbers: float(32bit), double(64bit)
    - Double-precision number: http://stackoverflow.com/questions/801117/whats-the-difference-between-a-single-precision-and-double-precision-floating-p
    - Round-off error: http://stackoverflow.com/questions/588004/is-floating-point-math-broken
    - What Every Computer Scientist Should Know About Floating-Point Arithmetic <http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html>

Eight-bit two's-complement integers

Bits	Unsigned value	Two's complement value
0111 1111	127	127
0111 1110	126	126
0000 0010	2	2
0000 0001	1	1
0000 0000	0	0
1111 1111	255	−1
1111 1110	254	−2
1000 0010	130	−126
1000 0001	129	−127
1000 0000	128	−128

Binary, Octal, Decimal and Hexadecimal numbers

- Conversion between them
    - Bin, Oct, Hex <=> decimal
    - **Hex <=> Bin (any hex digit expands to four bits)**
- Final numbers:
    - **`final double I_AM_A_FINAL_NUMBER_THAT_YOU_CANNOT_CHANGE`**

Operators

- Arithmetic operators

- Applies to: int and double
- Different type of operand
  - `3.0 /4`
  - `3 / 4`
  - `(int) 3.0 / 4`
  - `(double) (3/4)`
- `Rounding floating-point number to the nearest integer`
  - `Rounding positive floating …: (int) (x + 0.5)`
  - `Rounding negative floating …: (int) (x - 0.5)`

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo

- Relational operators
  - Applies to comparison of **primitive type, but floating-point number is an exception**
  - Used in: boolean expression
  - **== vs. =**

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>==	Greater than or equal to
<==	Less than or equal to

- Logical operators
  - Evaluated to : boolean
  - Short-circuit evaluation
    - `A||B when A == true, A && B when A == false, then B is not evaluated`
  - Truth table

P	Q	P && Q	P || Q	!P
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

!	NOT
&&	AND
||	OR

- Assignment

=	Simple assignment
+=, -=, *=, /=, %=	Compound assignment

- Increment and Decrement Operators
  - Diff between prefix(++i) and postfix(i++): side effect before/after evaluation

++	Increment
--	decrement

- Operator Precedence

Higher	!, ++, --
	*, /, %
	+, -
	<, >, <=, >=
	==, !=
	&&
	||
Lower	=, +=, -=, *=, /=, %=

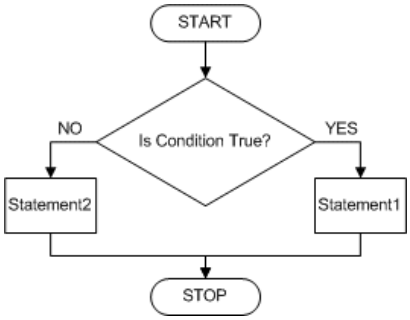
Input and output

- System.out is an object
  - `System.out.println()`
  - `System.out.print()`
- String concatenation
  - `System.out.println("There are " + numberOfPeople + " people");`
- Escape Sequences

\n	Newline
\"	Double quote
\\	backslash

Control structures

- Decision-making Control
  - `if`
  - `if-else`
  - Nested `if`
    - beware ***the dangling else***!
    - ```
      int n = IO.readInt();
      if (n > 0)
        if (n % 2 == 0)
          System.out.println(n);
      else
        System.out.println(n + " is not positive");
      ```
  - Extended `if` statement
- Iteration
  - For Loop
    - ```
      for(initialization; termination condition; update statement) {
        // body of loop
      }
      ```
  - For-each loop
    - ```
      for(SomeType element: collection) {
       // body of loop
      }
      ```
  - While loop
    - ```
      while (boolean test) {
        // body of loop
      }
      ```
  - Nested loop
    - A loop in another loop



Errors and exceptions