Math and Miscellaneous

Useful things that will be important sooner or later

Integer Division

- Integer division truncates the answer (cuts off the decimal)
- 3 / 5 → 0
- 3 / 5.0 → 0.6
- 3.0 / 5 → 0.6
- Be careful when dividing!

Type Casting

- Can use type casting to control the value type
- 3 / 5 → 0
- (double) 3 / 5 → 0.6
  - (double) 3 → 3.0
  - 3.0 / 5 → 0.6
- (int) 3.0 / 5 → 0
  - (int) 3.0 → 3
  - 3 / 5 → 0

Type Casting

- Type casting takes precedence over division
- (double) (3 / 5) → 0.0
  - 3 / 5 → 0
  - (double) 0 → 0.0
- (int) (3.0 / 5) → 0
  - 3.0 / 5 → 0.6
  - (int) 0.6 → 0

Math functions

- Math.abs(int x) → absolute value of x as an int
  - Math.abs(-5) → 5
- Math.abs(double x) → abs val of x as double
  - Math.abs(-3.7) → 3.7
- Math.pow(double base, double exp)
  → $base^{exp}$ as a double
  - Math.pow(3, 4) → 81.0
- Math.sqrt(double x) → square root of x
  - Math.sqrt(64) → 8.0
- Math.random()
  → random double in range[0.0, 1.0)
  - (int)(Math.random() * 20) → random integer in range [0, 19]

Misc Math functions/values

- Math.min(double a, double b)
  → returns the value which is smaller
- Math.max(double a, double b)
  → returns the value which is larger
- Math.PI → value of π

Error Types

- Syntax errors – errors that the compiler will catch
  - missing semicolons/curly braces, wrong data type, extra keywords, etc.
- Run-time errors – errors caught when you run your program (called Exceptions)
  - NullPointerException, ArrayIndexOutOfBoundsException, ArithmeticException, ClassCastException, IllegalArgumentException
- Logic errors – errors in your code that are legal but produce behavior that is not intended
  - Actual output is 0.79 instead of 79 (forgot to multiply by 100 for percentages)
- See Appendix D of Think Java for more info on errors

print vs println

- System.out.println() → prints to console and goes to next line
- System.out.print() → prints to console but stays on the same line
- System.out.print("Hello ");
  System.out.println("world!");
  System.out.println("APCSA");
  - Hello world!
    APCSA

Escape Characters

- Some String characters are interpreted as characters so escape characters allow us to indicate special meaning
- \\ → \
- \" → "
- \n → new line
  - System.out.print("\n"); is the same as System.out.println();

Comments

- /* */, // and /** */ are comments
- @param and @return are Javadoc comments (when creating a Javadoc for that method)
- Comments contain useful information about the method. If you're reading a method, read the comment. If you're writing a method, write a comment so other people can understand your method.