Important Processing Concepts (some review)

Processing Review
- The basic features of Processing
- Also Check Chapter 4 in the Reas & Fry book.
- Remember the Processing Reference Page: http://www.processing.org/reference/

Review: List of Items
- Variables and Declarations
- Assignments
- Expressions
- Repetition (looping) or For-statements
- Tests or if-statements
- Writing programs

Variables
- Facts about variables are ...
  - Variables “contain” their values, and they can be changed using assignment
  - Variables have a data type such as int, float, color, etc. which is the kind of data they contain
- Rules about variables are ...
  - Variables can be any string of letters, numbers or underscores (_) starting with a letter, case-sensitive
  - Variables must be declared; declarations at the top of the program or at the start of a function
  - Variables can be initialized in a declaration

Variables, the Picture
- Facts
  - "Contain their value": grade_point
  - "Assign to change": grade_point = 3.5
  - "Variables have data type":
- Rules
  - "Any string":
    - should mean something: ColorBlinky
    - theyAreUseful_4_U despite their_length
  - "Declare vars": int score; float gpa; color purple;
  - "Initializing is OK": int score=0; float gpa=4.0; color purple=color(128,0,128);

Assignments
- Facts about assignment:
  - Form is always <variable> = <expression>
  - Information moves from right to left
  - The <expression> is computed first, then the variable is changed, so x=x+1 is sensible
  - To exchange values in two variables takes 3 stmts
- Rules about assignment:
  - All assignment statements end with a semicolon
Assignments, The Picture

- Facts
  - "Form": grade_point=3.9; yellow=color(255,255,0);
  - grade_point is ILLEGAL
  - Info moves right to left": x = 4.0;
  - "Compute <expression> first": x = x + 1;
  - "Exchanging values of x, y takes 3 statements". Need to hold onto the value of x somewhere.

Expressions, The Picture

- Facts
  - Expressions are formulas using OPERATORS:
    - +, -, *, /, %, &&, ||, !=
  - Plus, minus, times, divide, mod, or, and, equals,
  - Operators can only be used with certain data types and their result is a certain data type
  - Can be smart to put in extra parentheses is OK
  - Self documenting code (variable names, parentheses, comments)

Expressions, Another Picture

- Facts
  - Expressions are formulas: a+b points*wgwt
  - (year%4 == 0) 7+4 (age>12) && (age<20)
  - Data types:
    - +, -, *, % want numbers
    - &&, || want logical (Boolean) values
    - == and != want arguments to be the same type
  - Parentheses are good: (a+b) * c is the same as a+b*c, but easier to read
  - Rules: Expressions replace vars": rect(x, y, x+4, y+4);

For loops (Repetition)

- Repeating commands is powerful:
  - Lightbot 2.0 used recursion, a function calling itself
  - Symbolic Lightbot prefixed a number, 2:Step

Expressions, the Picture

- Facts
  - Rules about expressions
    - Expressions can usually go anywhere that a variable can go
  - Can be smart to put in extra parentheses is OK
  - Operators can only be used with certain data types and their result is a certain data type
  - Can be smart to put in extra parentheses is OK
  - Self documenting code (variable names, parentheses, comments)

Expressions, Another Picture

- Facts
  - Rules about expressions
    - Expressions can usually go anywhere that a variable can go
  - Can be smart to put in extra parentheses is OK
  - Operators can only be used with certain data types and their result is a certain data type
  - Can be smart to put in extra parentheses is OK
  - Self documenting code (variable names, parentheses, comments)

Repetition, The Picture

- A for loop has several parts, all required ...
  - keyword open paren starting value continuation test loop close paren open brace
  - for (int i=0; i < 16; i++) {
    - if (i != 0) {
      - ellipses(100+25*i, 100, 15, 15);
    - }
  - }
  - The result of this statement is 16 copies of the stuff to be repeated. 16 Pacman pills

Repetition, Another Picture

- Or how about a bullseye?
  - Note the loop variable must be declared ... but could do it in loop itself like we did for pacman pills:
  - for (int i = 0; ...
If statements

- The instructions of a program are executed sequentially, one after another. Sometimes we want to skip some.
- We've used if in several HWs.
- If also has a required form
  ```
  if (year%4 == 0) {
    <stuff to do if condition true>
  }
  if (chosen_tint != red) {
    fill(chosen_tint);
  }
  ```

If Statements: the Picture

- An If-statement has a standard form
  - keyword
  - open paren
  - boolean expression
  - close paren
  - open brace
  - last, close brace

The result is that if bmi is in range (more than 18.5 and less than or equal to 24.9) then the fill color is green (indicating Ok)

Else Statement

- What happens if we want to do something else if the condition is false? What else? else!
- The else statement must follow an if ...
  ```
  if (year%4 == 0) {
    <stuff to do if condition true; if/Then Clause
  } else {
    <stuff to do if condition false; if/Else Clause
  }
  ```

Else, the Picture

- The standard form may now be obvious
  ```
  if (year%4 == 0) {
    feb_days = 29;
  } else {
    feb_days = 28;
  }
  ```
- Else must follow if because it does the test
- Finally, close brace

The result is that sets the number of days in February based on leap-year

If/Else, if else_red_blue

- Remember this early example?

Writing Programs

- Programs are given sequentially, the declarations at the top
- Braces {} are statement groupers ... they make a sequence of statements into just one thing, like the "true clause of an If-statement"
- All statements must end with a semicolon, except the grouping braces ... they don't end with a semicolon
- Name your variables something meaningful
- Generally white space doesn't matter; comment your code