if Statement

• A condition is an expression that can be either true, or false.
• “if” statement allows a program to make a decision based on the value of a condition.
Equality Operators

• ==
  e.g. \( x == y \) means \( x \) is equal to \( y \)

• !=
  e.g. \( x != y \) means \( x \) is not equal to \( y \)
Relational Operators

- `>` means “greater than”
- `<` means “less than”
- `>=` means “greater than or equal to”
- `<=` means “less than or equal to”

e.g.

- `x>y`
- `c<d`
- `e>=f`
- `g<=k`
Rule of Operator Precedence

• All equality operators are of same precedence.
• All Relational operators are of same precedence.
• Equality operators are lower in operator precedence than relational operators.
• Relational operators are lower in operator precedence than arithmetic operators.
• Assignment operator is lower in operator precedence than equality operator
• Operator Precedence rule of thumb: When in doubt, add parenthesis!
Rule of Operator Precedence

• * / %
• + -
• < <= > >=
• == !=
• =
Common Programming Errors

- Forget left and/or right parenthesis for the condition in “if” statement.
- Confusing an equality operator, “==”, with an assignment operator, “=”
- ==, !=, <=, >= These operators cannot contain spaces between symbols like “= =”
- Reverse the operator !=, <=, >=, as in “=!, =<, =>
- Place a semicolon immediately after the right parenthesis of condition in an “if” statement
Algorithms

- A computer program of a finite set of well-defined instructions for solving a problem.
- A set of actions (statements) & the order in which they execute.
- Program control specifies the order in which these statements execute.
Algorithm of Wizard’s Valentine Date

- Get out of bed
- Take off pajamas
- Take a shower
- Get dressed
- Eat breakfast
- Drive to meet my date
Algorithm of Nutty’s Valentine Date

• Get out of bed
• Take off pajamas
• Get dressed
• Take a shower
• Eat breakfast
• Drive to meet my date
Order matters

• The order of statement execution matters.
• The order of execution are controlled by control structures
Java Control Structure

- **Sequence Structure.** By default, JVM executes Java statement in the order they are written.

- **Selection Statements.** Selectively execute some statements depending on the value of conditions. “if”, “if…else”, “switch”

- **Repetition Statements.** Looping Statement. Program performs statements repetitively as long as certain conditions remain true. “for”, “while”, “do…while”.
if Statement

- If the condition (boolean) is true, perform the action, otherwise, skip it.
- if (grade >= 60.0)
  System.out.println("You have passed!");
- if (grade >= 60)
  {
    // use block code in place of a single statement
    System.out.println("You have passed!");
    System.out.println("Congratulations!");
  }
if .. else Statement

- If the condition is true, perform the first statement(or block), otherwise, perform the second statement(or block).
- if (haveTicket) //when haveTicket == true
  
  ```java
  System.out.println("Watch game at the Stadium!");
  System.out.println("I am lucky!");
  ```

else //when haveTicket != true

```java
    System.out.println("Watch TV at home.");
    System.out.println("I am okay.");
}
```
Nested if .. else Statement

- if .. else can be nested to have multiple selection.
- If (temperature > 85)
  System.out.println(“It is hot.”);
else if (temperature<68)
  System.out.println(“It is cold.”);
else
  System.out.println(“It is cool.”);
Dangling else problem

• Java compiler always associates an “else” with the immediately preceding “if” unless told otherwise by using {}.

• Block code is java statements enclosed within {}. A block code can be placed anywhere in a program that a single statement can be placed.
Example

• int l = 6;
  int m = 3;
  if (l > 5)
    if (m > 5)
      System.out.println("Both l & m are greater than 5.");
    else
      System.out.println("l>5 & m<5");
  else
    System.out.println("l<=5"); //We don’t know “m” yet at this point.
• if (l > 5)
  {
    if (m > 5)
      System.out.println("Both l & m are greater than 5.");
  }
Conditional Operator

- Condition ? Do_if_true : do_if_false
- Can be used in place of “if .. else” statement
- It takes three operands. The first operand is the boolean expression. The second operand is the conditional expression if the boolean expression is true. The third operand is the conditional expression if the boolean expression is false.
- System.out.println(grade>=60 ? “passed” : “failed”);
Reading Assignment

• “Java 2”. Finish chapter 2. P71-76 of Chapter 3