CS102
Introduction to data structures, algorithms, and object-oriented programming

DAY 4
printf, in the System.out class, allows you to specify the number of places after the decimal point in a double. The printf method’s first argument is a String that contains a *format string*. The format string is embedded in the String argument as `%1.2f` as shown below and in Section 2.4.1 of our textbook.

```
System.out.printf("Your BMI is %1.2f.\n", bMI);
```

The `\n` embeds a newline after the String is printed. The `,` separates the String with embedded format characters from the variable to be embedded in the String.
Scanner, Sect. 2.4.6

Scanner is a class in the java.util package that provides another way of taking input from the keyboard. To use a Scanner to read from the keyboard (aka, standard input), you need to instantiate an object of type Scanner.

```
Scanner in = new Scanner(System.in);
double rnum = in.nextDouble();
```

This is the first time we’ve used the keyword “new”. new is used to create a new object of type Scanner. The input is done by calling an instance method of the new Scanner object.
Control Structures (Ch. 3)

1. block {....}
2. while loop*
3. do while loop*
4. for loop*
5. if decision maker* (branching statement)
6. switch decision maker* (branching statement)
Blocks

Statements can be grouped by enclosing them in {}s, a block.

Blocks can contain any number of statements, including 0.

As a matter of good programming style, you should lay out your program on the page in a way that will make its structure as clear as possible. In general, this means putting one statement per line and using indentation to indicate statements that are contained inside a block.

Blocks are generally associated with other structures: class bodies, method bodies, and the other control structures. Variables declared inside a block are only visible within that block.
While loop

Used to repeat a given statement over and over (even infinitely)

A while loop will repeat a statement over and over, but only so long as a specified condition remains true. A while loop has the form:

```
while (boolean-expression) {
    statements
}
```

1. boolean-expression (b-e) is evaluated
2. if b-e is true, evaluate inner block
3. if b-e is false, start evaluating statements after block
It is possible to write a while loop that stops somewhere in the middle (and often this type is easier to understand).

/* Get a positive integer from the user. */

int N; // declare N
while (true) {
    System.out.print("Enter a positive integer: ");
    N = TextIO.getlnInt();
    if (N > 0) {
        break;
    }
    System.out.println("That number is not positive. Please try again.");
}
if

1. if ...else: either or type statement, each with its own block of code.

2. if alone with a block of code.

3. if, else if, else if, ..., else. Multi-decision statements, each with its own block of code.

if and else combinations are the closest thing Java has to the cond statement in Racket. Only one clause in the group is executed and the rest are ignored. The else at the end is like that in the cond, sort of a default condition.
while & if flow of control diagrams

while

if
If..Else Flow of Control

Is condition true?

Yes

Do statement 1

No

Do statement 2

if

else
for loops

The initialization is done inside the parenthesis at the beginning of the loop.

A loop control variable is declared, initialized, tested, and modified in the for loop statement.
Nested loops

Often used in coding matrices but also in many other instances.

```java
String str;  // Line of text entered by the user.
int count;   // Number of different letters found in str.
char letter; // A letter of the alphabet.

System.out.println("Please type in a line of text.");
str = TextIO.getln(); // call to static getln methods in class TextIO
str = str.toUpperCase(); // call on non-static method in object str
count = 0;   // initialize count
System.out.println("Your input contains the following letters:");
System.out.println();

for ( letter = 'A'; letter <= 'Z'; letter++ )
{
    int i; // Position of a character in str.
    for ( i = 0; i < str.length(); i++ ) {
        if ( letter == str.charAt(i) ) {
            System.out.print(letter);
            System.out.print(' ');
            count++;
            break;
        }
    }
}
```

Algorithms

Step-by-step description of how to solve a problem.

Each line of human language must be broken down into a language solvable by a computer.

Developing a program from a human language form involves what is called stepwise refinement. That is, re-write each line into a form called pseudocode and then write it in a computer language.