• Reminder: Assignment 1 is due Monday, 11 September, 11:59 pm

• Reading for next class is LDC Chapter 5

• SI sessions and drop-in hours starting!
  • Drop-In Hours Sundays: 5-7, 8-10
  • SI Sundays: 7-8 (Priscilla)
  • SI Mondays: 6-7 (Ainsley)
  • Drop-In Hours Mondays: 7-10
  • SI Tuesday: 7-8 (Priscilla)
  • SI Thursday: 6-7 (Ainsley)
  • Drop-In Hours Thursdays: 7-8

• Java has a boolean type that can take the value true or false
  • Booleans arise naturally when using relational operators to compare two values
    
    \[
    \begin{align*}
    3 < 5 & \quad \text{false} \\
    3 < 2 & \quad \text{true} \\
    3 > 2 & \quad \text{false} \\
    5 <= 1 & \quad \text{false} \\
    5 >= 1 & \quad \text{true} \\
    5 == 5 & \quad \text{true} \\
    5 == 6 & \quad \text{false} \\
    5 != 6 & \quad \text{true}
    \end{align*}
    \]

• Boolean values can be manipulated with the logical operators ! (not), && (and), and || (or)
  
  \[
  \begin{align*}
  ! (3 < 5) & \quad \text{true} \\
  ! (3 == 5) & \quad \text{false} \\
  (3 > 5) && (7 < 8) & \quad \text{false} \\
  (3 < 5) && (7 < 8) & \quad \text{true} \\
  (3 > 8) || (7 < 8) & \quad \text{true} \\
  (3 > 8) || (7 > 8) & \quad \text{true}
  \end{align*}
  \]
A predicate is any method that returns a boolean value

```java
//determine if n is even
public static boolean isEven(int n) {
    return (n % 2) == 0;
}
```

```java
//determine if num is divisible by factor
public static boolean isDivisibleBy(int num, int factor) {
    return (num % factor) == 0;
}
```

```java
//determine if n is between lo and hi
public static boolean isBetween(double n, double lo, double hi) {
    return (lo <= n) && (n <= hi);
}
```

Write your own predicate to determine if n is odd. Then, can you write it another way?

```java
public static boolean isOdd(int n) {
    return !isEven(n);
}
```

To choose between two courses of action, to control the program flow, we use conditional statements such as if, else if, and else.

```java
//returns absolute value of n
public static double abs(double n) {
    if (n < 0) {
        return -n;
    } else {
        return n;
    }
}
```

```java
public void main(String[] args) {
    int x = 28; String s = "meow";
    if (x < 30 && s.length() < 10) {
        x = x + 5;
        int y = s.length();
        if (x+y > 36) {
            System.out.println("hello " + x);
        } else if (x+y < 33) {
            System.out.println("howdy " + y);
        } else {
            System.out.println("hi!");
        }
    } else {
        x = x - 10;
        int y = s.length() + 5;
        if (x == 15) System.out.println("Salut " + x);
        else System.out.println("Ciao " + y);
    }
}
```
while

- **Iteration** refers to a sequence of steps that is repeated until some stopping condition is reached.

```java
while(boolean_expression){
    statement 1;
    statement 2;
    ...
}
```

1. Evaluate boolean expression
2. If true, execute body of loop and go back to step 1
3. If false, go to statement after while loop

```java
int i = 1;
while (i < 4) {
    System.out.println("CS230");
    i = i + 1;
}
```

for

```java
for ( ; ; ) {
    statements
}
```

- Execute this statement once before entering loop
- If true, execute body of loop
- Execute this statement before next test of the boolean expression
- If the boolean expression was true, the body of the loop will be executed
- If the boolean expression evaluates to false, drop down to here

```java
for (int i = 1; i < 4; i++) {
    System.out.println("CS230");
}
```

Math Class

```java
System.out.println(Math.max(100, 50));
System.out.println(Math.sqrt(25));
System.out.println(Math.log(10));
```

// Given area of circle, returns the circle's radius.
// Since area = pi * r^2, we have r = squareRoot(area / pi).
public static double getCircleRadius(double area) {
    return Math.sqrt(area / Math.PI);
}

```java
System.out.println(getCircleRadius(100));
```

Random Class

```java
import java.util.Random;

public class RandomExample {
    public static void main(String[] args) {
        Random rand = new Random();
        for (int i = 0; i < 15; i++) {
            System.out.println(rand.nextInt(10));
        }
    }
}
```
// Returns true if character is lower-case
// vowel (a, e, i, o, u), false otherwise.
public static boolean isVowel(char ch)

// Returns the number of occurrences of
// vowels in the String s
public static int countVowels(String s)