Lectures 5&6
Strings

- Sequence of characters
- Reference type (non-primitive)
- Specified by double quotes ("")
- Can have length 0 – empty string = ""

Examples:
- String name = "Dan";
- String className = "CS302: Intro to Programming";
STRING OPERATIONS

- Concatenation (+)
  - Have already seen in our output statements
  - Ex: String name = "Ned" + " Stark";
  - String className = "cs";
  - int classNum = 302;
  - className = className + classNum; //className is now: "cs302"

- Length
  - String name = "Luke Skywalker";
  - int length = name.length(); //length = 14
    - Remember identifier.methodName()
CHARS

- Single character
- Specified by single quotes (')
- Has numeric value
- Ex.

```java
char myChar = 'a';
System.out.println(myChar);  //will print out: a
myChar++;
System.out.println(myChar);  //will print out: b
```
int x = (int)'a';
System.out.println(x); //output: 97
char myChar = (char)(x++);
System.out.println(myChar); //output: b
CHARAT

- Method to find a specific character within a String
- Strings are 0-indexed
- Ex.
  - String name = "Dan Szafir";
  - char first = name.charAt(0); //first = 'D'
  - int length = name.length(); //length = ?
  - char last = name.charAt(length - 1); //last = 'r'
- What if I had done:
  - char last = name.charAt(length);
What if I want to get part of a String?

String name.substring([start], [end])
  - Will include charAt(start)
  - Will include charAt(end - 1);
  - Will NOT include charAt(end)
  - Start, end, must be ints

Remember the 0-indexed nature of Strings

Ex.

String name = "Dan Szafir";
String first = name.substring(0, 3);
String last = name.substring(4);
INDEXOF

- Opposite of charAt
- Finds the first occurrence of a char in a String

```java
System.out.println("Enter your favorite team");
String name = in.nextLine(); //Assume Boston Bruins was entered
int spaceIndex = name.indexOf(' '); //spaceIndex = 6
String city = name.substring(0, spaceIndex);
```

- Will return -1 if the specified character was NOT in the String
STRING METHODS SUMMARY

- `length()` - counts the number of chars in a String
- `charAt([index])` – returns the char at [index]
- `substring([start], [end])` – returns a String whose content is the character at [start] up to but not including the char at [end]
- `substring([start])` – returns a String whose content is the character at [start] through the end of the original String
- `indexOf([char])` – returns the first occurrence of [char] in the String, or -1 if it wasn't found
Switch to Eclipse
What if I want to make a decision?

Parts:

- **Boolean expression** (a statement that is either true or false)
- **Code**

**Ex.**

```java
if (5 > 1) {
    System.out.println("Five is greater than 1");
}
```
COMPARING NUMBERS: RELATIONAL OPERATIONS

- **==**
  - Is something equal to something else
  - if (a == b)
- **>**
  - Greater than
- **<**
  - Less than
- **>=**
  - Greater than or equal to
- **<=**
  - Less than or equal to
- **!=**
  - Not equal

Precedence

Lower precedence than arithmetic operators

Ex. what does (3 + 2 < 5) evaluate to?
COMPARING STRINGS

- Do NOT use ==
- Strings are reference variables, not primitives
- Instead use .equals() and .equalsIgnoreCase()
- Also .compareTo()
  - Returns an int
- Format:
  - stringOne.equals(stringTwo)

```java
String foo = "abcdef";
String bar = "ABCDEF";
if (foo.equals(bar))
{
    System.out.println("foo equals bar");
}
if (foo.equalsIgnoreCase(bar))
{
    System.out.println("foo equals bar if you ignore the case");
}
```
Else

- Code that executes if the boolean expression was false

![Flowchart diagram]

- Boolean Expression
- Code under if block
- Code under else block
- Continue execution of code
String foo = "abcdef";
String bar = "ABCDEF";
if (foo.equals(bar))
{
    System.out.println("foo equals bar");
}
else
{
    System.out.println("foo doesn't equal bar");
}
**ELSE IF**

- What if I wanted to check more than one thing?
  - if (test something)
    - {
      //do something
    }
  - else if (test something else)
    - {
      //do something else
    }
  - else
    - {
      //what to do if both of those tests were false
    }
IF, ELSE IF, AND ELSE

- If
  - One or none
- If...Else
  - One or another
- If...Else If...Else
  - One of many
    - The only thing you need is an if
      - Can have if and else ifs with no else
      - Can have if and else with no else ifs
      - Can have if alone
      - CANNOT have and else if or an else without a starting if
SCOPE

- Determines in what context values and expressions are associated

General Rule of Thumb:

- Variables defined within a set of braces are only good within that set (and any nested sets)
PRACTICE 1

Write a simple Log-In program:

Input: Username

Output:

- If username matches a known username output: "Hello [username], good to see you again!"
- Else: "Invalid Login Attempt"
Write a program according to these specifications:

- **Input:** Day of the week, Year
- **Output:**
  - If Sunday: “Yikes, tomorrow I have to work again :(
  - If Saturday: “Hooray, I can hang out with friends today :)
  - If Monday: “Alas, I’m lecturing right now~~~ but cheer up"
  - Otherwise: "Just another weekday, let’s enjoy working"
  - If Year is evenly divisible by 4: "Leap year, we can all live 1 more day this year, isn’t that great?"
Write a program to convert numerical grades to letter grades:

- Input: a numerical grade 0 – 100
- Output:
  - Grade: 90 - 100: "A, you must have a great IQ score"
  - Grade: 80 – 90: "B, it's okay, but just don't tell your mom"
  - Grade 70 – 80: "C, got addicted to Diablo?"
  - Grade 60 – 70: "D, oops, the student passes out"
  - Grade 0 – 60: "F, no matter what, your instructor is just ruthless"
  - Otherwise: "Error, invalid grade"
PRACTICE 4: MULTI-PLANET WEIGHT CONVERTER
Alternative to having if...else if... else if... else...

Use if you are testing the same variable in each boolean expression

Ex.

```java
int day = in.nextInt(); String dayName = "";
switch(day)
{
    case 1: dayName = "Sunday"; break; //same as if (day == 1)
    case 2: dayName = "Monday"; break; //else if (day == 2)
    case 3: dayName = "Tuesday"; break;
    //...more cases...
    default: dayName = ""; break; //default is a catchall like else statements
}
Will "fall through" to the next case if there is no break:

```java
switch(day)
{
    case 1: dayName = "Sunday"; break;
    case 2: dayName = "Monday"; break;
    case 3: dayName = "Tuesday"; break;
    //...more cases...
    default: dayName = ""; break;
}
```

What is the value of dayName if day = 1?
Sometimes you want to fall through

```java
switch(day)
{
    case 1:
    case 7:
        dayType = "weekend"; break;
    case 2:
    case 3:
    case 4:
    case 5:
    case 6:
        dayType = "weekday"; break;
    default:
        dayType = "unknown"; break;
}
```
PRACTICE 5

- Use switch statement
- Take an int input for the nth day in a week (starting from Sunday)
  - If (Sat or Sun), print out “It's a weekend”
  - If (MonTuTh), print out “It's a weekday”
  - If (Wed), print out “Halfway there”
  - If (Fri), print out “It's almost the weekend”
  - Otherwise, print out “Not a valid day (day's should be 1-7)”
NESTED BRANCHES

- If statement within another if, else if, or else statement

```python
if (I'm hungry)
{
    if (I feel like Italian)
    {
        Make spaghetti
    }
    else
    {
        Make a hamburger
    }
}
```
int temperature = in.nextInt();
int raining = in.nextInt(); //1 means yes, otherwise no
if (temperature > 70) {
    if (raining == 1) {
        S.o.pln("Wear shorts and bring an umbrella");
    }
    else {
        S.o.pln("Wear shorts and sunglasses");
    }
}
else {
    S.o.pln("It is indeed a typical WI weather");
}
BOOLEAN VARIABLES

- Booleans are either true or false;
- Ex.
  boolean failed = false;
  if (failed) //same as if (failed == true)
  {
    //stop the program
  }
I want an apple and an orange
I want an apple or an orange
BOOLEAN OPERATORS

- Ways to combine boolean variables
  - && = and
  - || = or
  - ! = not
VISUALIZING BOOLEAN OPERATORS

college || university

poverty && crime

cats && !dogs
"I carry an umbrella if it rains or snows"
   if (rain || snow) : carry umbrella

"I only wear shorts if its hot and sunny"
   if (hot && sunny) : wear shorts
if (I'm hungry)
{
  if (I feel like Italian)
  {
    Make spaghetti
  }
  else
  {
    Make a hamburger
  }
}

if (I'm hungry && I feel like Italian)
{
  Make spaghetti
}
else if (I'm hungry)
{
  Make a hamburger
}
Wrong

if (solved)
{}

else
{
   //lots of code
}

Right

if (!solved)
{
   //lots of code
}

//if(!solved) is the same
//as if (solved == false)
INPUT VALIDATION

- Use if statements to make sure the user input a valid value
- `in.hasNextInt()` - check if the user input an int
- `in.hasNextDouble()` – check if user input a double
More complicated Login Program

Input: Username, Password

If username matches a known username and password matches the corresponding password, output: *Giddy Up!*

Otherwise output: *Invalid Login*

Known Usernames / Passwords:

- Jerry / porsche
- George / Bosco