

## Part II - Brent Wenerstrom

- Compiling multiple files
- Constructors
- Main

# Compile Multiple Files

- How do we compile the FavoriteMovie and another class?

# Compile Multiple Files Example

- Explicit way: list out every file for `javac`

```
$ javac FavoriteMovie.java FavoriteMovieRunner.java
```

-> FavoriteMovie.class, FavoriteMovieRunner.class

- Now we can run `FavoriteMovieRunner` class through it's `main` method.

```
$ java FavoriteMovieRunner
```

# Constructors

- Setup a class
- Entry point when using the keyword **new**

# Constructor Example

```
class Halloween
{
    String date;

    // Constructor
    Halloween()
    {
        date = "Oct 31st";
    }
}
```

```
Halloween halloween = new Halloween();
```

# Parameter Passing

- Provide a "mailbox"
- Enable passing of values between different parts of code
- Example:

```
void double(int x) {  
    return x + x;  
}
```

```
int y = double(3);  
int z = double(y);
```

# Constructor with Parameters

- Each instance can have their own "personality"

```
public class Actor
{
    String name;

    // Constructor with parameter n
    public Actor(String n)
    {
        name = n;
    }

    public void speak()
    {
        System.out.println("Hi! My name is " + name);
    }
}
```

## Constructor with Parameters (cont)

Create Actor instances and use them.

```
Actor freeman = new Actor("Morgan Freeman");  
Actor hanks = new Actor("Tom Hanks");  
  
freeman.speak();  
hanks.speak();
```



# Constructor with Parameters (output)

```
$ java ActorExample  
Hi! My name is Morgan Freeman  
Hi! My name is Tom Hanks
```

- We could execute the same method and get different output
- This came because each instance held different values (personality)

# Main method missing

```
$ java Actor
Error: Main method not found in class Actor, please
define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend
javafx.application.Application
```

- A class without a main cannot be run
- Main is built into the language and requires a precise definition

# Main method

And now we can execute the Actor class

```
class Actor
{
    String name;
    Actor(String n)
    {
        name = n;
    }

    public static void main(String[] args)
    {
        System.out.println("Actor class");
    }
}
```

```
$ java Actor
Actor class
```

# Homework Discussion

- Let's get started on the homework
- Due Friday Oct. 6
- Make best effort and turn it in