The Java Memory Model

- There are a variety of things to discuss to learn about the entire Java memory model.
- In this course we are going to focus on:
  - variables, references to objects, and objects
  - stack versus heap
Stack and Heap

- The stack and the heap are two areas within memory where information is stored.
- In a typical Java program in this course
  - the stack can be seen as the part of memory that has sections assigned to certain variables and references right when we launch a program
  - the heap can be seen as the part of memory that has sections dynamically assigned to hold certain variables, references, and objects as they are created, and which can also have assigned slots freed for reuse

Student

Let's consider the following lines of code as if it were in the main method of our program and see what is placed on the stack and what is placed on the heap.

```java
int val = 7;
Student s1 = new Student("Pat", val, 987654321);
Student s2 = s1;
```
Garbage Collection

• The short version: the parts of memory on the heap that are no longer referenced by anything are available to be released for reuse.

Consider the previous picture modified to reflect the additional line of code shown here:

```java
int val = 7;
Student s1 = new Student("Pat", val, 987654321);
Student s2 = s1;
s1 = new Student("Sam", 3, 436389456);
```

Is anything on the heap ready for garbage collection?
Consider the previous picture modified to reflect the additional line of code shown here:

```java
int val = 7;
Student s1 = new Student("Pat", val, 987654321);
Student s2 = s1;
s1 = new Student("Sam", 3, 436389456);
s2 = new Student("George", 5, 965456746);
```

Is anything on the heap ready for garbage collection?
