CS 61B: Lecture 6 Code

When you create an array of objects, Java does not create the objects automatically. The array contains space for references to the objects. You must create the objects yourself.

```
String[] sentence = new String[3];
sentence[0] = "Word";
sentence[2] = new String();
```

Reading command-line arguments sent to your Java program.

```
class Echo {
  public static void main(String[] args) {
    for (int i = 0; i < args.length; i++) {
      System.out.println(args[i]);
    }
  }
}
```

A "do" loop.

```
do {
  s = keybd.readLine();
  process(s);
} while (s.length() > 0);              // Exit loop if s is an empty String.
```

Two "time-and-a-half" loops, equivalent to each other.

```
s = keybd.readLine();                |  while (true) {       // Loop forever.
while (s.length() > 0) {             |    s = keybd.readLine();
  process(s);                        |    if (s.length() == 0) {
  s = keybd.readLine();              |      break;
}                                    |    }
                                     |    process(s);
                                     |  }
```

Two equivalent loops with more than one natural endpoint. (The loop at right is written without the "break" statement, but it's harder to understand.)

```
for (int i = 0; i < 10; i++) {       |  int i = 0;
  s = keybd.readLine();              |  do {
  if (s.length() == 0) {             |    s = keybd.readLine();
    break;                           |    if (s.length() > 0) {
  }                                  |      process(s);
  process(s);                        |    }
}                                    |    i++;
                                     |  } while ((i < 10) &&
                                     |     (s.length() > 0));
```

What's the difference between the following two loops?

```
int i = 0;                           | for (int i = 0; i < 10; i++) {
while (i < 10) {                     |   if (condition(i)) {
  if (condition(i)) {                |     continue;
    continue;                        |   }
  }                                  |   call(i);
  call(i);                           | }
  i++;                               |
}                                    |
```