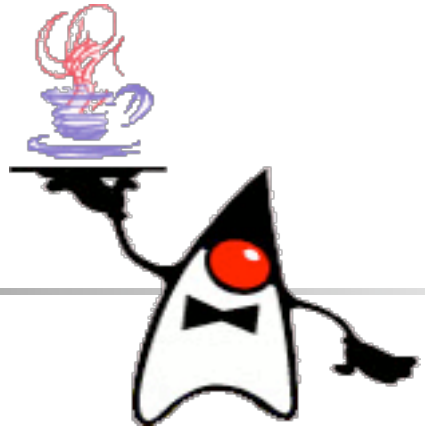
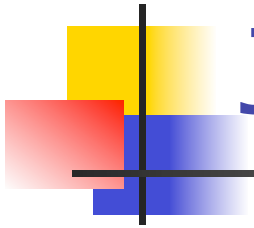
CSCI 143 Software Eng I

Rhys Price Jones

On to Java

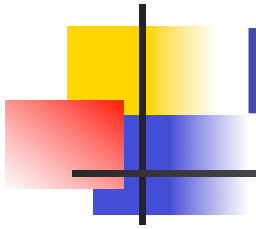


- Hello World
- the boring way
- a better way
- see the lab exercises



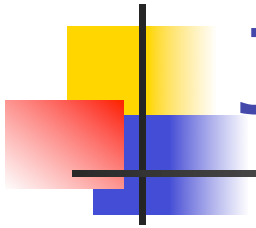
Java is

- object oriented
  - create classes
  - inherit from parent classes
  - create objects
- compiled or interpreted?
  - look at HelloBoring.class
  - look at executable C
  - do it now!



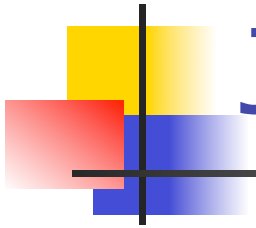
Byte Code

- Intermediate code
- Interpreted by JVM
  - Java Virtual Machine
- Foo.java is compiled
  - giving Foo.class
- The bytecode in Foo.class
  - is interpreted by JVM



Java has

- a runtime environment supporting
  - GUIs
  - threads
  - networking etc.
- a large library
- development support
  - including javadoc

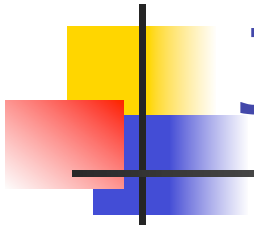


JDK

- Download Java, you get the Java Development Kit (JDK) containing
  - javac -- the compiler
  - java -- the interpreter
  - appletviewer -- for viewing applets
  - support tools including
    - javadoc
    - jar

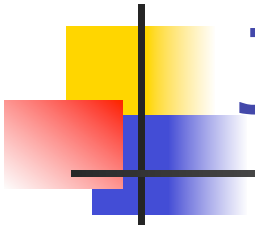
History

- James Gosling, Sun MicroSystems
- Oak (1991)
- HotJava 1995
- Netscape built Java 1.02 into browser
- Major versions
  - 1.0 (dead and buried)
  - 1.1 changed event model
  - 1.2 1.3 1.4
  - 1.5 generics



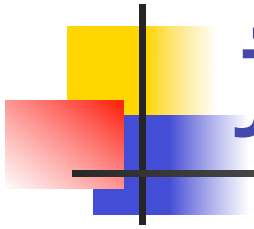
Java is simple

- My javac is 56K
- My java is 52K
- My classes.jar is 21.3M



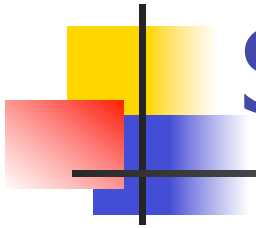
Java is secure

- no direct access (via pointers) to memory
- interpreter checks bytecodes before execution
- sandbox model
- code can be digitally signed



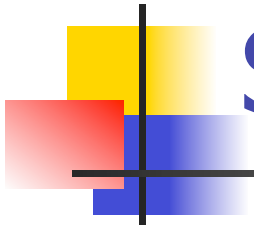
java is portable

- any machine with a JVM can run it
- any browser implementing JVM can run Java applets
- data types are standardized
  - an int is 4 bytes
  - a char is 2 bytes
    - ??!!??
- but...



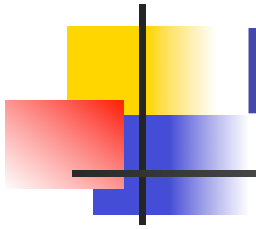
Strings

- You met strings and the concatenation operator in the somewhat less boring program.
  - System.out.print(”Hello “ + name + “. How are you?”);
- In Java, strings are objects, of the class java.lang.String
- You do not need to import java.lang.*



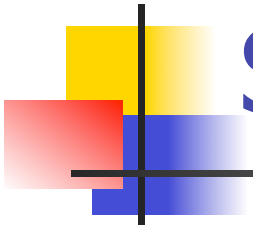
Strings are objects.

- Strings are not primitive types!
- Strings are immutable.
- You can create new strings in the "object style":
  - String foo = new String("I am a string");
- or in the "shorthand style":
  - String goo = "I am also a string";



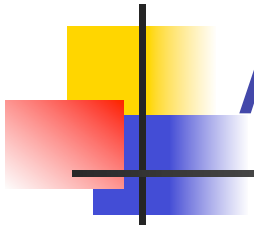
Docs for java.lang.String

- Java provides an extensive set of methods for manipulating strings.



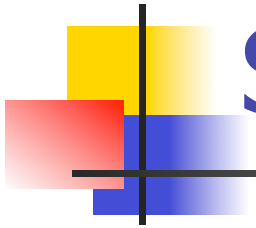
String Equality

- Strings are objects.
- In the declaration String name;
- name is a reference, not the string itself.
  - String name1 = new String (“Joe”);
  - String name2 = new String (“Joe”);
- Is (name1 == name2)?
- == determines object equality



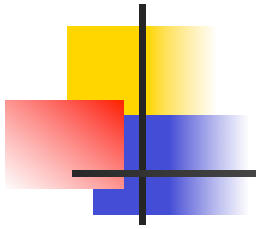
A Common Mistake

- A common mistake
- You can’t determine string equality using ==.
- So how?



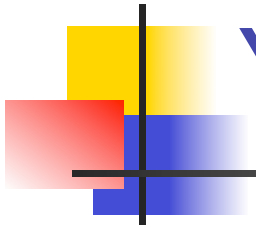
Strings are immutable.

- To create strings you can mutate (change), you must declare StringBuffer objects.



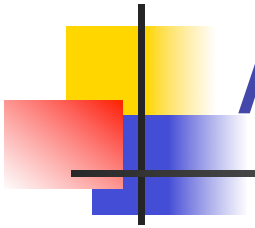
String concatenation

- The + sign is overloaded in Java. We have already seen where it can mean:
  - addition 5 + 7
  - increment count++
- + can also indicate concatenation.
  - “abc” + “xyz” results in “abcxyz”
  - “123” + “456” results in … what?
- Concatenation
  - System.out.println



You've already met arrays

- public static void main (String[] args)
- What is an array?
  - An array is a data structure.
  - An array can hold many items.
  - All items in an array must be of the same type.
  - Arrays have bounds which determine the valid indices of the array

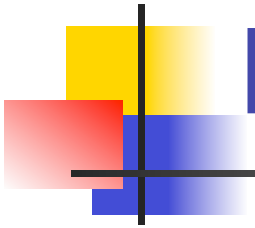


Arrays in Java

- As with other data types, an array must be declared. Create the array using the new operator.

```
int myIntArray[] = new int
 [25];
```

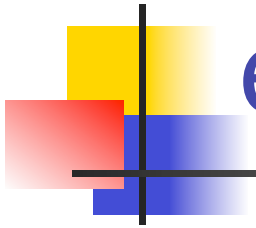
  - This declares an array of 25 integers called `myIntArray`.
  - Not more than 25 integers can be contained in this array.



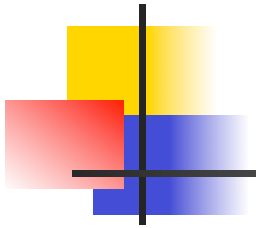
Problems with arrays

- An array's size is fixed upon creation.
- If you try to reference an invalid index in an array, you will cause a fatal run-time exception to be thrown (this is very bad).
- Any attempt to access an invalid index results in an array out of bounds exception.

Some other languages do not enforce this discipline

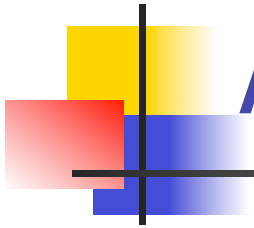


- A major weakness allowing for hacker attacks
- Java was designed with network safety a paramount consideration.



What does this code do?

```
int myIntArray[] = new int [5];
myIntArray[0] = 0;
myIntArray[1] = 0;
myIntArray[2] = 0;
myIntArray[3] = 0;
myIntArray[4] = 0;
```

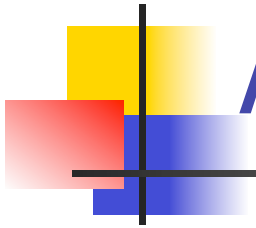


Actually Nothing!!

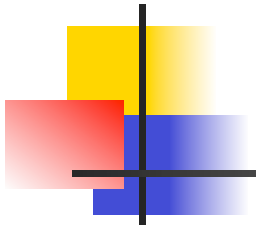
- By default, Java initializes every element of an int array to 0
- How do you know that?

```
public class CheckArray {
  public static void main(String[] args) {
  int[] foo = new int[20];
  int i;
  for (i=0; i<10;; i++)    // initialize just the first 10 elements
    foo[i] = i;
  System.out.println("Filled  10 spots of an array of size 20”);
  System.out.println(“Accepted default initializations of the rest”);
  for (i=0; i<20; i++) System.out.println(foo[i]);
  }
}
```

Other Ways to Instantiate Arrays



- `int[] smallPrimes = {2, 3, 5, 7, 11, 13, 17, 19};`
- `String[] days = {”sunday”, ”monday”, “tuesday”, “wednesday”, “thursday”, “friday”, “saturday”};`
- `static final String[] suits = {"clubs", "diamonds", "hearts", "spades"};`
- `static final String[] ranks = {"ace", "deuce", "trey", "four", "five", "six", "seven", "eight", "nine", "ten", "knave", "queen", "king"};`



Short digression

- `Math.random()`
- is a method in the Math class that returns a pseudo-random double in the range [0..1)
  - that means including 0, but not 1
- (int) (10 * Math.random()) returns a random integer 0,1,..,9