Comparing Strings

- Need to be quite sure what we are comparing:

```
String s = ”Hello”;
String t = ”Hello”;
...
if (s == t)
   ...
```

- Compares the *references*
- Are they equal?

Comparing Strings (cont.)

- Yes, they are in this case:

- The compiler tries to minimise the creation of automatic **`String`** objects

Comparing Strings (cont.)

- Safer to use the **`equals`** method:

```
String s = "Hello";
String t = new String("Hello");
...
if (s.equals(t))
   ...
```

Chapter 9
Exception Handling

Exceptions

- Used for almost all error-handling in Java:
  - Programming errors
  - User/environmental errors
  - System errors

Example

```
46  int[] a = new int[3];
47  System.out.println(a[5]);
```

```
java.lang.ArrayIndexOutOfBoundsException
     at Test.main(Test.java:47)
Exception in thread "main"
```

Types of Exception

Errors

- Fatal system errors
  - e.g. JVM out of memory
  - Usually ignored by programs

Runtime Exceptions

- Programming errors
  - e.g. Array subscript out of range, division by zero
  - Can be prevented by careful programming
  - May be ignored by programs

Other Exceptions

- All remaining exceptions
  - e.g. file not found, network errors, security breaches
  - The program *may* be able to recover from these
  - The program *cannot* ignore the possibility that they may arise
    - *catch* or *throw*

Catching Exceptions

```
int[] a = new int[3];
try
  { System.out.println(a[5]);
  }
catch (ArrayIndexOutOfBoundsException e)
  { System.err.println("ERROR: " + e);
  }
```

- Not a good example!

Catching Exceptions (cont.)

- The catch is given the exception (an object)
  - Like a parameter

```
catch (ArrayIndexOutOfBoundsException e)
  { System.err.println("ERROR: " + e);
  }
```

- Here we simply print it:

```
ERROR: java.lang.ArrayIndexOutOfBoundsException
```

Catching Exceptions (cont.)

- Can “ignore” the exception:

```
catch (SomeException e)
  {  // Do nothing
  }
```

Catching Multiple Exceptions

- Order of catches is important
  - Specific exceptions first
  - More general ones later

```
try
  { ...
  }
catch (FileNotFoundException e)
  { System.err.println("File not found");
  }
catch (IOException e)
  { System.err.println("I/O ERROR: " + e);
  }
```

Miscellaneous Issues

- Error messages
  - Use **`System.err`**
- Terminating the program
  - Use **`System.exit(1)`**