

Programming
Exceptionally



David Evans
www.cs.virginia.edu/cs205

Quiz Answers

2. What is an object?

- Java-specific answers:
 - What you get when you invoke a class constructor
 - An instance of a class
- General answers
 - An entity that includes both *state* and *procedures* for manipulating that state
- Really general answers
 - "Something intelligible or perceptible by the mind." (Philosophy dictionary answer)

Mystery Method

```
public void f (String s) {  
    char c = s.charAt (0);  
    if (c == '-') {  
        s.concat ("negative");  
    }  
}
```

```
public void f (String s) {  
    char c = s.charAt (0);  
    if (c == '-') {  
        s.concat ("negative");  
    }  
}
```

```
public char charAt(int index)  
    // REQUIRES: The value of index is  
    // between 0 and the length of this - 1.  
    // EFFECTS: ...
```

```
public String concat(String s)  
    // EFFECTS: Returns a new string that is  
    // the concatenation of this followed by s.
```

Mode Specification

```
public int mode (int [] a)  
    // MODIFIES: a  
    // EFFECTS: Returns the value that  
    // appears most often in a.
```

Mode Specification

```
public int mode (int [] a)  
    // MODIFIES: a  
    // EFFECTS: Returns the value that  
    // appears most often in a.
```

If something is listed in MODIFIES, how it can change must be described in EFFECTS. Recall that MODIFIES means everything *not listed* is unchanged.

Mode Specification

```
public int mode (int [] a)
// REQUIRES: a has at least one element
// EFFECTS: Returns the value that
// appears most often in a.
```

Note this is misleading.
There may be multiple values.

cs205: engineering software

7

Implementing Mode

```
static public int mode (int[] a) {
    int best = a[0];
    int bestcount = 1;
    for (int i = 0; i < a.length; i++) {
        int val = a[i];
        int count = 0;
        for (int j = 0; j < a.length; j++) {
            if (a[j] == val) { count++; }
        }
        if (count > bestcount) {
            best = val; bestcount = count;
        }
    }
    return best; }
}
```

Note: I am using poor code formatting to fit on one slide. Your code should not look like this! (Hint: use Ctrl-Shift-F in Eclipse)

cs205: engineering software

8

Violating Requires

- In C/C++: can lead to anything
 - Machine crash
 - Security compromise
 - Strange results
- In Java: *often* leads to runtime exception

cs205: engineering software

9

```
static public int mode (int[] a) {
    int best = a[0];
    int bestcount = 1;
    for (int i = 0; i < a.length; i++) {
        int val = a[i];
        int count = 0;
        for (int j = 0; j < a.length; j++) {
            if (a[j] == val) { count++; }
        }
        if (count > bestcount) {
            best = val; bestcount = count;
        }
    }
    return best; }
}
```

Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 0
at Quiz.mode(Quiz.java:3)
at Quiz.main(Quiz.java:27)

cs205: engineering software

10

Use Exceptions to Remove Requires

```
static public int mode (int [] a)
// REQUIRES: a has at least one element
// EFFECTS: Returns the value that
// appears most often in a.
```



```
static public int mode (int [] a)
throws NoModeException
// EFFECTS: If a is empty throws NoModeException.
// Otherwise, returns the value that appears most
// often in a.
```

cs205: engineering software

11

Throwing Exceptions

```
static public int mode (int [] a) throws NoModeException
{
    if (a == null || size () == 0)
        throw new NoModeException ();
    ...
}
```

What is `NoModeException`?

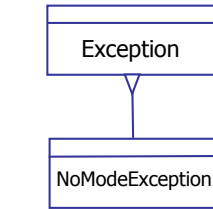
cs205: engineering software

12

Exceptions are Objects

```
public class NoModeException
  extends Exception
{
  public NoModeException () {
    super ();
  }
}
```

extends Exception means
EmptyException inherits from the
Exception type (in the Java API).



We will cover subtyping and inheritance later.

cs205: engineering software

13

Compiler Checking

```
static public void main(String[] args) {
  int[] tarray1 = { 1, 2, 2, 3, 2, 5 };
  int[] tarray2 = {};

  System.out.println("Mode tarray1: " + mode(tarray1));
  System.out.println("Mode tarray2: " + mode(tarray2));
}
```

Unhandled exception type NoModeException

cs205: engineering software

14

Catching Exceptions

```
static public void main(String[] args) {
  int[] tarray1 = { 1, 2, 2, 3, 2, 5 };
  int[] tarray2 = {};

  try {
    System.out.println("Mode tarray1: " + mode(tarray1));
  } catch (NoModeException nme) {
    System.err.println("Error: " + nme);
  }

  try {
    System.out.println("Mode tarray2: " + mode(tarray2));
  } catch (NoModeException nme) {
    System.err.println("Error: " + nme);
  }
}
```

Code inside the try block executes normally until it throws an exception. If no exception is thrown, execution proceeds after the catch. If the NoModeException exception is thrown, the catch handler runs.

cs205: engineering software

15

Charge

- PS2 is due Friday
- Next class:
 - Lots more issues with Exceptions
 - Data abstraction

cs205: engineering software

16