

COS 126 Precept

Will Clarkson
February 24, 2009

Agenda

- ▶ Exam I Info
- ▶ Functions
- ▶ Activity 1
- ▶ Pass by value/reference
- ▶ Overloading
- ▶ Passing Arrays
- ▶ Activity 2



Exam 1

- ▶ **When: March 10th**
- ▶ **Where: TBA (at same time as class and precept)**
- ▶ **Two Parts**
 - ▶ In Class Exam: Written Questions
 - ▶ Programming: Mini-Assignment
- ▶ **What you need**
 - ▶ In Class: Just your mind
 - ▶ Precept: Bring a laptop if possible, or you can use the computers in Friend 017



Functions

- ▶ You have used them before, you just didn't know it
 - ▶ `Math.sqrt(x)`, `Math.pow(a,b)` are both functions
- ▶ Can be used to do more than just mathematical formulas
- ▶ Break up program into smaller, more manageable chunks
 - ▶ Allows individual testing/debugging
- ▶ **Syntax:**
`public static <return type> functionName(Type1 arg1, Type2 arg2, ..., TypeN argN)`



Functions

- `public static <return type> functionName(Type1 arg1, ..., TypeN argN)`
 - Use 'void' as return type if we don't want to return anything.
 - i.e. `System.out.println();`
- `main` is a static method, just like any other
 - `public static void main(String [] args)`
- Call static methods by using name of class
 - StdDraw class has public function named `picture`
 - `StdDraw.picture(x, y, filename)`
- Public means other classes can call this method



Pass By Value (Lets send a Copy)

- ▶ When we pass primitive types (int, double, boolean) as arguments, we actually send a copy
 - ▶ It is as if we declared a new variable, and assigned the argument to it before we start executing the function



Activity 1

- ▶ Write method:

```
public static int max3(int a, int b, int c)
```



Can two functions have the same name?

- ▶ **Yes!**

- ▶ Provided that they take different input arguments (type or number)

- ▶ `int max(int a, int b, int c)`

- ▶ `double max(double a, double b, double c)`

- ▶ This wouldn't work

- ▶ `void max(int a, int b, int c)`

- ▶ Because arguments are same, and return type is the only difference

- ▶ This is called **Overloading** a function



Activity

- ▶ Write method `max3` that takes doubles instead of integers



Pass by Reference (when we pass arrays)

- ▶ If array is large, copying it takes a lot of time
 - ▶ Slows down computer program a lot
 - ▶ So, arrays are passed by value (no copying is done)
 - ▶ If you modify an array inside a function then that change affects the array outside of the function.
- ▶ Lets see an example:
 - ▶ Reversing an array

```
public static _____ reverse(_____);
```
 - ▶ Notice we don't need a return type, all changes to a are reflected in calling function.
- ▶ We can return a new array if we want though
 - ▶ We don't ALWAYS have to modify the input array if we choose not to



Returning an Array

- ▶ Create version of reverse that returns an array
 - ▶ `public static Return Type [] functionName(inputs.....)`

