

Java Programs

You have to create a class!

You run the class.

- for this to work, the class must have a method named `main()` that is declared as

```
public static void main( Strings[] arg)
```

- if you don't do this right, you see something like this when you try to run a class:

```
Exception in thread "main" java.lang.NoSuchMethodError: main
```

The Java Language

1

`public static void main`

public: This method can be accessed (called) from outside the class.

static: This method does not require that an object of the class exists. static methods are sort-of like "global" methods, they are always available (but you have to use the class name to get at them).

void: no return value.

The Java Language

2

`main(Strings[] arg)`

- `main()` will be passed an array of Strings corresponding to any command line parameter values.

- If you ran a program (class) like this:

```
java Foo hi there dave
```

- then `Foo.main()` would be passed an array (with length 3) of String objects. The values would be "hi", "there" and "dave".

`arg[0]` is the String "hi", ...

The Java Language

3

Show Command Line Program

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

The Java Language

4

Converting **String** to **int**

```
int x = Integer.parseInt("12");
```

Integer is actually the name of a *class*.

parseInt() is one of the methods of class **Integer**.

The details are in the Java API documentation in `java.lang`, class **Integer**

The Java Language

5

Printing the command line args as ints

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

The Java Language

6

Issues

What happens if the user doesn't type in a bunch of integer on the command line?

```
java ShowArgs hello there
```

`Integer.parseInt` throws an exception.

- we didn't catch the exception, the program is terminated and some default exception handling code is executed (a stack trace is printed).

The Java Language

7

Catching the Exception

```
public class ShowArgs {
    public static void main(String[] args) {
        try {
            for (int i=0;i<args.length;i++)
                System.out.println("args["+i+"] = "
                    + Integer.parseInt(args[i]));
        } catch (NumberFormatException e) {
            System.out.println("Error: I need ints!");
        }
    }
}
```

The Java Language

8

Another Method

```
public class ShowArgs {
    public static void main(String[] args) {
        for (int i=0;i<args.length;i++)
            try {
                System.out.println("args["+i+"] = "
                    + Integer.parseInt(args[i]));
            } catch (NumberFormatException e) {
                System.out.println("Skipping (not an int)");
            }
    }
}
```

The Java Language

9

Homework !

You need to write a few classes.

- they are each independent.
- they each need a main()
- they use command line arguments to determine what they do.
- you will need to look through the Java API docs to figure out how to do some things...
