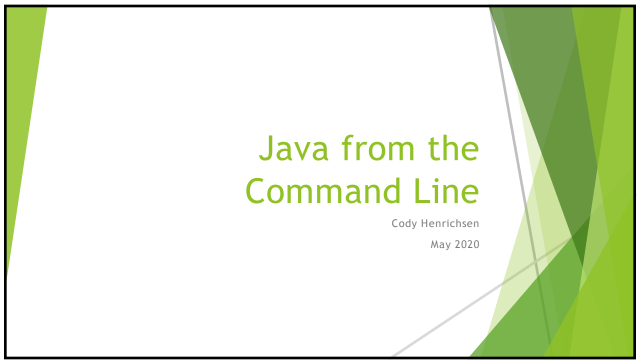


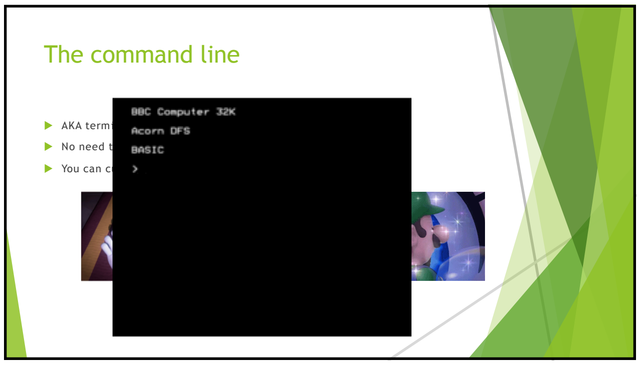
Java from the Command Line

Cody Henrichsen

May 2020

Vocab

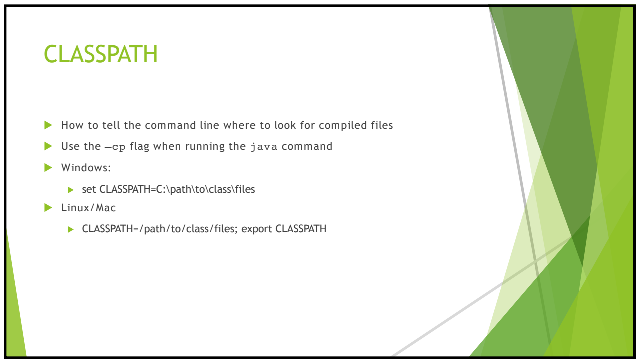
- CLASSPATH
- Command Line
- flags
- java
- javac



The command line

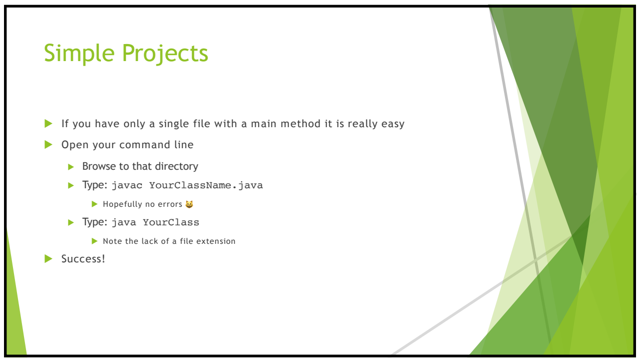
- AKA term
- No need t
- You can c

```
BBC Computer 32K
Acorn DFS
BASIC
>
```



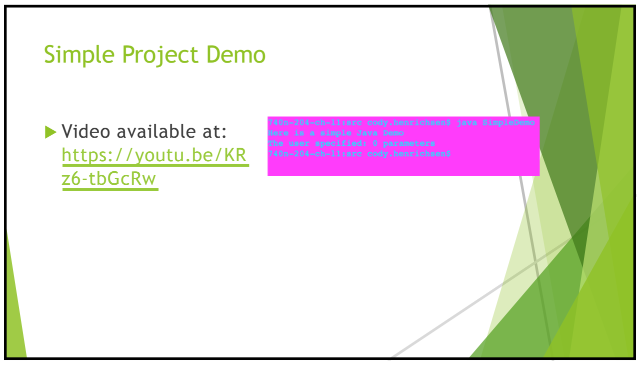
CLASSPATH

- How to tell the command line where to look for compiled files
- Use the `–cp` flag when running the `java` command
- Windows:
  - set CLASSPATH=C:\path\to\class\files
- Linux/Mac
  - CLASSPATH=/path/to/class/files; export CLASSPATH



Simple Projects

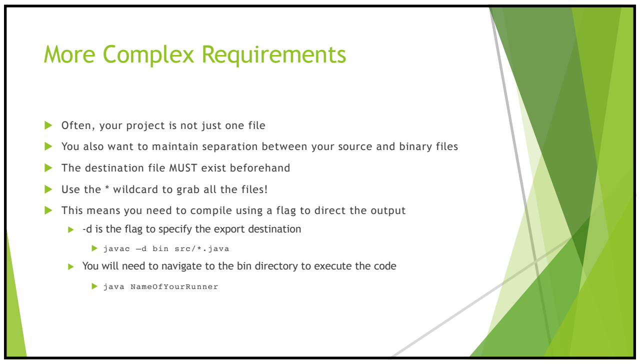
- If you have only a single file with a main method it is really easy
- Open your command line
  - Browse to that directory
  - Type: `javac YourClassName.java`
    - Hopefully no errors 😸
  - Type: `java YourClass`
    - Note the lack of a file extension
- Success!



Simple Project Demo

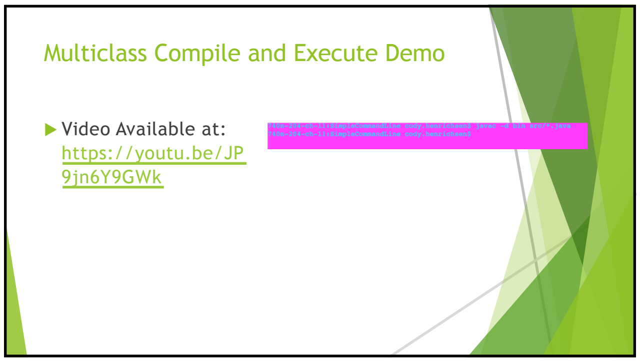
- Video available at: https://youtu.be/KRz6-tbGcRw

```
740n-204-ch-11:src cody.henrichsen$ java SimpleDemo
Here is a simple Java Demo
The user specified: 0 parameters
740n-204-ch-11:src cody.henrichsen$
```



More Complex Requirements

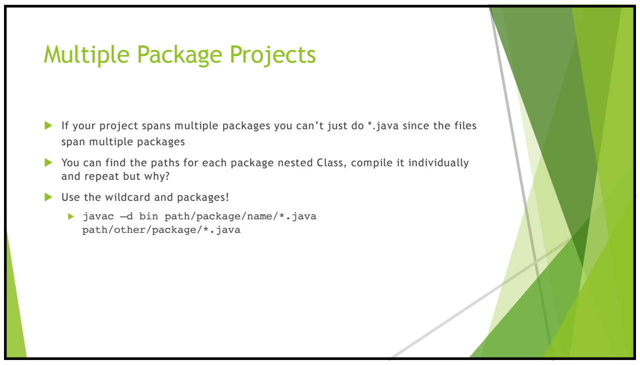
- Often, your project is not just one file
- You also want to maintain separation between your source and binary files
- The destination file MUST exist beforehand
- Use the * wildcard to grab all the files!
- This means you need to compile using a flag to direct the output
  - -d is the flag to specify the export destination
    - `javac –d bin src/*.java`
  - You will need to navigate to the bin directory to execute the code
    - `java NameOfYourRunner`



Multiclass Compile and Execute Demo

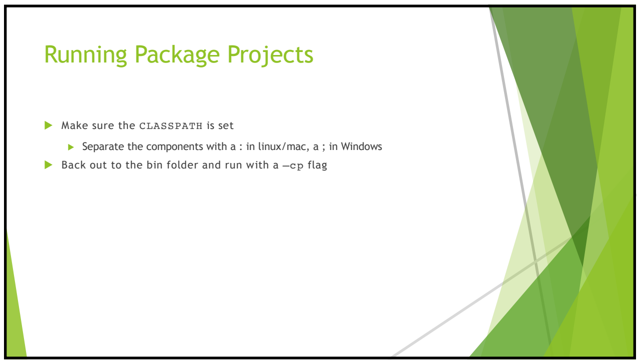
- Video Available at: https://youtu.be/JP9jn6Y9GWk

```
740n-204-ch-11:SimpleCommandLine cody.henrichsen$ javac -d bin src/*.java
740n-204-ch-11:SimpleCommandLine cody.henrichsen$
```



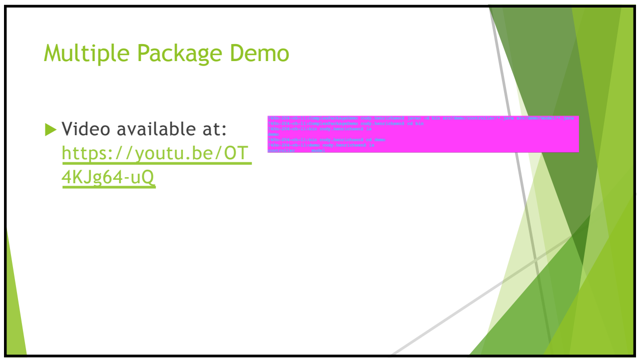
Multiple Package Projects

- If your project spans multiple packages you can't just do *.java since the files span multiple packages
- You can find the paths for each package nested Class, compile it individually and repeat but why?
- Use the wildcard and packages!
  - `javac –d bin path/package/name/*.java path/other/package/*.java`



Running Package Projects

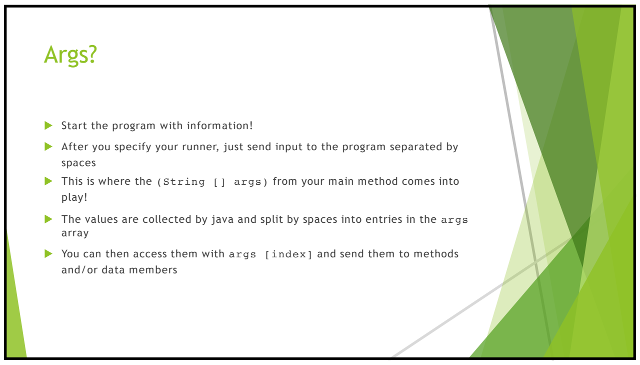
- Make sure the `CLASSPATH` is set
  - Separate the components with a : in linux/mac, a ; in Windows
- Back out to the bin folder and run with a `–cp` flag



Multiple Package Demo

- Video available at: https://youtu.be/OT4KJg64-uQ

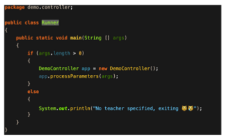
```
740n-204-ch-11:ComplexPackageDemo cody.henrichsen$ javac -d bin src/demo/controller/*.java src/demo/model/*.java
740n-204-ch-11:ComplexPackageDemo cody.henrichsen$ cd bin
740n-204-ch-11:bin cody.henrichsen$ ls
demo
740n-204-ch-11:bin cody.henrichsen$ cd demo
740n-204-ch-11:demo cody.henrichsen$ ls
controller      model
```



Args?

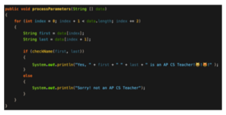
- Start the program with information!
- After you specify your runner, just send input to the program separated by spaces
- This is where the `(String [] args)` from your main method comes into play!
- The values are collected by java and split by spaces into entries in the `args` array
- You can then access them with `args [index]` and send them to methods and/or data members

Args Sample



```
package demo.controller;

public class Runner
{
    public static void main(String [] args)
    {
        if (args.length > 0)
        {
            DemoController app = new DemoController();
            app.processParameters(args);
        }
        else
        {
            System.out.println("No teacher specified, exiting 😺😿");
        }
    }
}
```

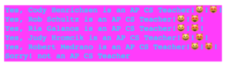


```
public void processParameters(String [] data)
{
    for (int index = 0; index + 1 < data.length; index += 2)
    {
        String first = data[index];
        String last = data[index + 1];

        if (checkName(first, last))
        {
            System.out.println("Yes, " + first + " " + last + " " is an AP CS Teacher!😺!😻!" );
        }
        else
        {
            System.out.println("Sorry! not an AP CS Teacher");
        }
    }
}
```

Args Demo

- Video available at: https://youtu.be/zD7ilPCcZrM



```
Yes, Cody Henrichsen is an AP CS Teacher!😺!😻!
Yes, Rob Schultz is an AP CS Teacher!😺!😻!
Yes, Ria Galanos is an AP CS Teacher!😺!😻!
Yes, Judy Hromcik is an AP CS Teacher!😺!😻!
Yes, Robert Medrano is an AP CS Teacher!😺!😻!
Sorry! not an AP CS Teacher
```

Review

- Compile projects with javac file(s)
  - -d flag specifies where to put the compiled files
- Run programs with java RunnerName
  - -cp flag to reference CLASSPATH
- Send data to the program with Strings
  - java MyRunner Send this info as parameters