Java Syntax and Output

Lab 0A

A bare bones class!

```
public class CompSci
{



}
```

All Java programs start with a class.

bare bones + a main!

```
public class CompSci
{
   public static void main(String[] args)
   {
      System.out.println("Comp Sci!");
   }
}
```

OUTPUT

```
Comp Sci!
```

syntax rules

```
public class CompSci
{  //open brace
   public static void main(String[] args)
   {
      System.out.println("Comp Sci!");
   }
}  //close brace
```

Braces – You gotta have 'em!   Every class and every method must have a { and a } .

syntax rules

```
public class CompSci
{
   public static void main(String[] args)
   {
      System.out.println("Comp Sci!");
   }
}
```

You must put a semi-colon at the end of all Java program statements (;).

{ and ; rule

Never put a ;
before an open { brace

```
;{  //illegal
};  //legal
```

indentation

```
public class CompSci
{
   public static void main(String[] args)
   {
      System.out.println("Comp Sci!");
   }
}
```

Indent all code 3 spaces to make it easier to read.

Open compsci.java

Java Output

System.out

frequently used methods

Name	Use
print(x)	print x and stay on the current line
println(x)	print x and move to next line down
printf(s,x)	print x according to s specifications

Java Output

reference

command / method

```
System.out.print("compsci");
```

OUTPUT

```
compsci
```

Java Output

```
System.out.print("compsci");
System.out.print("compsci");
```

OUTPUT

```
compscicompsci
```

Java Output

```
System.out.println("compsci");
```

OUTPUT

```
compsci
```

Java Output

```
System.out.println("compsci");
System.out.println("compsci");
```

OUTPUT

```
compsci
compsci
```

Open one.java

Java Output

Escape Sequences

\n	newline
\t	tab
\r	carriage return
\b	backspace

```
System.out.println("c\tompsci");
```

OUTPUT

```
c       ompsci
```

Java Output

Escape Sequences

\n	newline
\t	tab
\r	carriage return
\b	backspace

```
System.out.println("com\tpsci");
```

OUTPUT

```
com	psci
```

Java Output

Escape Sequences

\n	newline
\t	tab
\r	carriage return
\b	backspace

```
System.out.println("comp\nsci");
```

OUTPUT

```
comp
sci
```

Open two.java

Java Output

Escape Sequences

\\	outs \
\"	outs "
\’	outs ’

```
System.out.println("\\compsci\"/");
```

OUTPUT

```
\compsci"/
```

Java Output

Escape Sequences

\\	outs \
\"	outs "
\’	outs ’

```
System.out.println("\\'comp\'sci\'/");
```

OUTPUT

```
\'comp'sci'/
```

Escape Sequences

frequently used combinations

Name	Use
\t	tabs over five spaces
\n	moves to front of next line
\b	deletes previous character
\r	moves to front of current line
\\	nets one backslash \
\"	nets one double quote "
\’	nets one single quote ’

Java Comments

//	**single-line comments**
/* */	**block comments**

```
//this line prints stuff on the screen
System.out.println("stuff");
```

Java Comments

//	single-line comments
/* */	block comments

```
/*
   this line prints stuff on the screen
*/
System.out.println("stuff");
```

Java Output

```
System.out.printf("%s","compsci\n");
```

OUTPUT

```
compsci
```

Open three.java

Errors

Syntax errors occur when you type something in wrong, causing the code to not compile.

```
//missing semicolon - ; expected
System.out.println("stuff")

//case problem – should be System
system.out.println("stuff")
```

Errors

Runtime errors occur when something goes wrong while the program is running.

```
//an out of bounds exception is thrown
String s = "runtime_error";
System.out.println( s.charAt(15) );
```

Open four.java

Start work on Lab 0a