CPSC 319 Tutorial 03
Techniques for 1st Assignment

Longsheng Zhou

Department of Computer Science
University of Calgary

January 26, 2015

A Tour

Tuesday, Jan.26

1. Java Basics: “hello world”, etc.
2. Java Output: Console and .txt File.
3. Parse Command Line Arguments.

Thursday, Feb.02

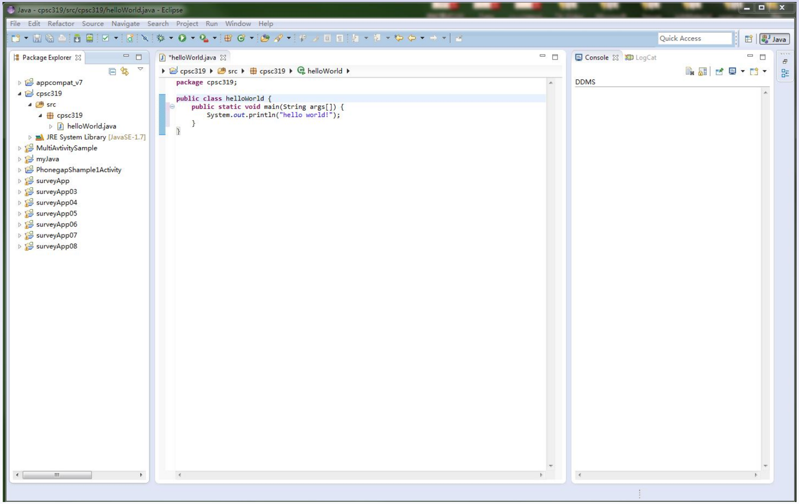
1. Create an Array with Arbitrary length and Random Integer.
2. Time Section of Code.
3. (Maybe) Bubble Sort and Insertion Sort.

Choosing Your Development Environment

- Integrated Development Environment(IDE): Eclipse, etc.
  Q: How to configure a java project in Eclipse?
  A: `https://www.youtube.com/watch?v=DJMIWwyvVAM`
- Plain Text: command-line tools (Wins, Linux, etc).

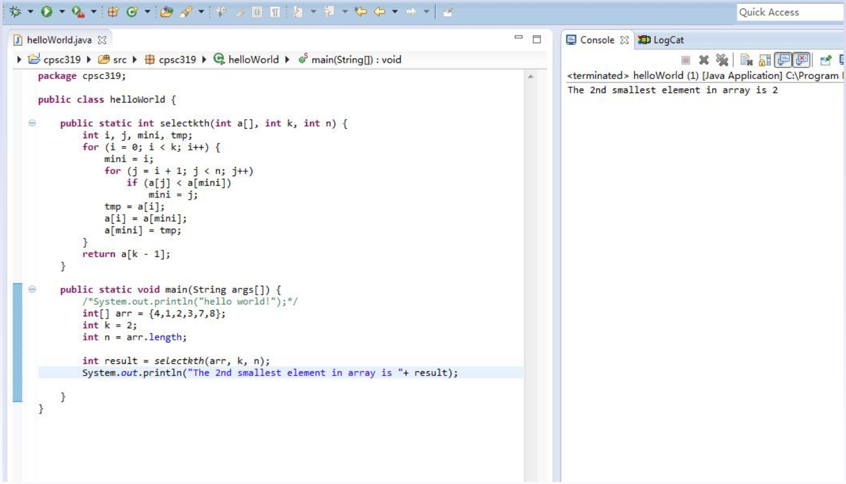
ps: For both cases, make sure you have installed the java JDK(java development kit) first. In the following tutorials, we will mainly use Eclipse and also introduce some command line techniques.

1st java program via Eclipse:



```
package cpsc319;

public class helloWorld {
    public static void main(String args[]) {
        System.out.println("hello world!");
    }
}
```

2nd java program via Eclipse:

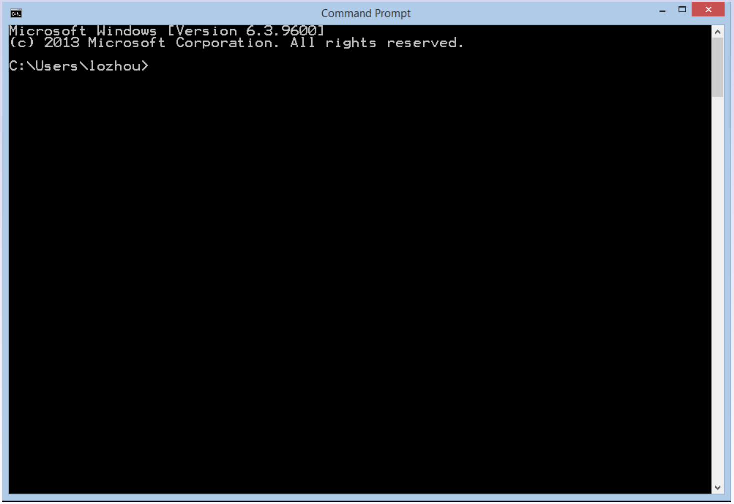


```
package cpsc319;

public class helloWorld {

    public static int selectkth(int a[], int k, int n) {
        int i, j, mini, tmp;
        for (i = 0; i < k; i++) {
            mini = i;
            for (j = i + 1; j < n; j++)
                if (a[j] < a[mini])
                    mini = j;
            tmp = a[i];
            a[i] = a[mini];
            a[mini] = tmp;
        }
        return a[k - 1];
    }

    public static void main(String args[]) {
        /*System.out.println("hello world!");*/
        int[] arr = {4,1,2,3,7,8};
        int k = 2;
        int n = arr.length;

        int result = selectkth(arr, k, n);
        System.out.println("The 2nd smallest element in array is "+ result);

    }
}
```

```
<terminated> helloWorld (1) [Java Application] C:\Program
The 2nd smallest element in array is 2
```

Reference: *Drozdek* Textbook 2nd edition page 77.

How to run .java by command line?



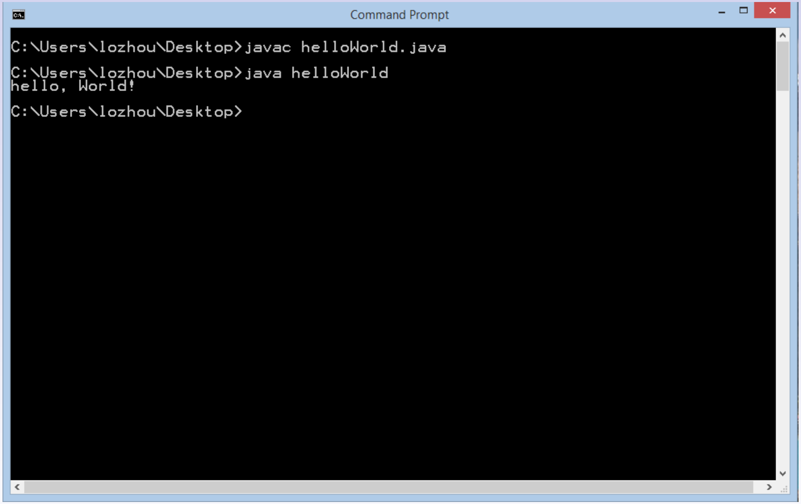
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\lozhou>
```

Figure: Windows "Command Prompt (cmd)"

Some useful commands:

- *cd*
- *dir*
- *cls*
- *javac*
- *java*

ps: When you first time execute cmd to run .java file. You would have several problems. Be patient and find the solution online. e.g. If you use "javac" command and get an error like "Javac is not recognized as an internal or external command" then this reference could help you.
https://www.youtube.com/watch?v=jerx_0b-qRY



```
C:\Users\lozhou\Desktop>javac helloWorld.java

C:\Users\lozhou\Desktop>java helloWorld
hello, World!

C:\Users\lozhou\Desktop>
```

Two Options:

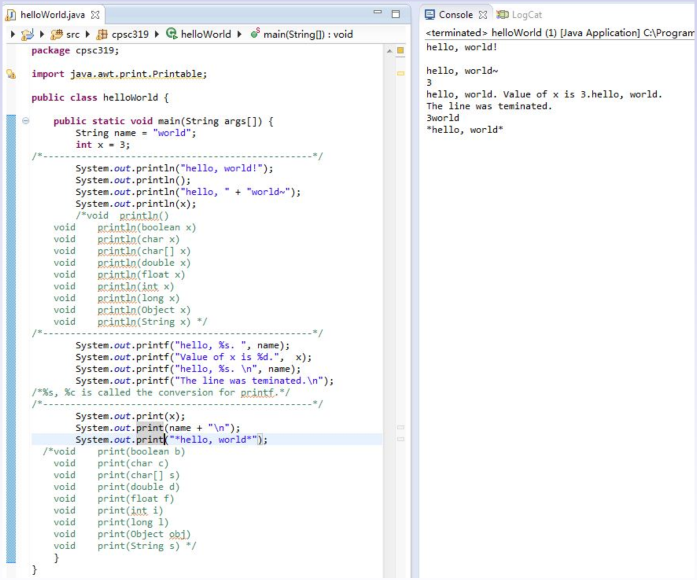
- Console
  Shown in the 1st and 2st program.
- .txt (text) file.

Three Options:

- System.out.println(“hello, ” + “world!”);
- System.out.print(x);
- System.out.printf(“hello, %s.”, name);

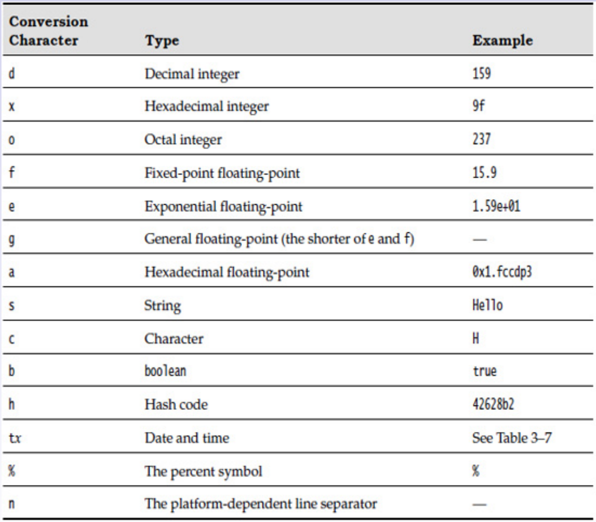
More details?

`https://docs.oracle.com/javase/6/docs/api/java/io/PrintStream.html`



```java
package cpsc319;

import java.awt.print.Printable;

public class helloWorld {

    public static void main(String args[]) {
        String name = "world";
        int x = 3;
/*-------------------------------------------------*/
        System.out.println("hello, world!");
        System.out.println();
        System.out.println("hello, " + "world~");
        System.out.println(x);
        /*void  println()
    void    println(boolean x)
    void    println(char x)
    void    println(char[] x)
    void    println(double x)
    void    println(float x)
    void    println(int x)
    void    println(long x)
    void    println(Object x)
    void    println(String x) */
/*-------------------------------------------------*/
        System.out.printf("hello, %s. ", name);
        System.out.printf("Value of x is %d.",  x);
        System.out.printf("hello, %s. \n", name);
        System.out.printf("The line was teminated.\n");
/*%s, %c is called the conversion for printf.*/
/*-------------------------------------------------*/
        System.out.print(x);
        System.out.print(name + "\n");
        System.out.print("*hello, world*");
  /*void    print(boolean b)
    void    print(char c)
    void    print(char[] s)
    void    print(double d)
    void    print(float f)
    void    print(int i)
    void    print(long l)
    void    print(Object obj)
    void    print(String s) */
    }
}
```

```
<terminated> helloWorld (1) [Java Application] C:\Program
hello, world!

hello, world~
3
hello, world. Value of x is 3.hello, world.
The line was teminated.
3world
*hello, world*
```



Conversion Character	Type	Example
d	Decimal integer	159
x	Hexadecimal integer	9f
o	Octal integer	237
f	Fixed-point floating-point	15.9
e	Exponential floating-point	1.59e+01
g	General floating-point (the shorter of e and f)	—
a	Hexadecimal floating-point	0x1.fccdp3
s	String	Hello
c	Character	H
b	boolean	true
h	Hash code	42628b2
tx	Date and time	See Table 3–7
%	The percent symbol	%
n	The platform-dependent line separator	—

- To write to a file(.txt), construct a ***PrintWriter*** object. In the constructor, simply supply the file name or the path:

```
PrintWriter out = new PrintWriter(“myFile.txt”);
```

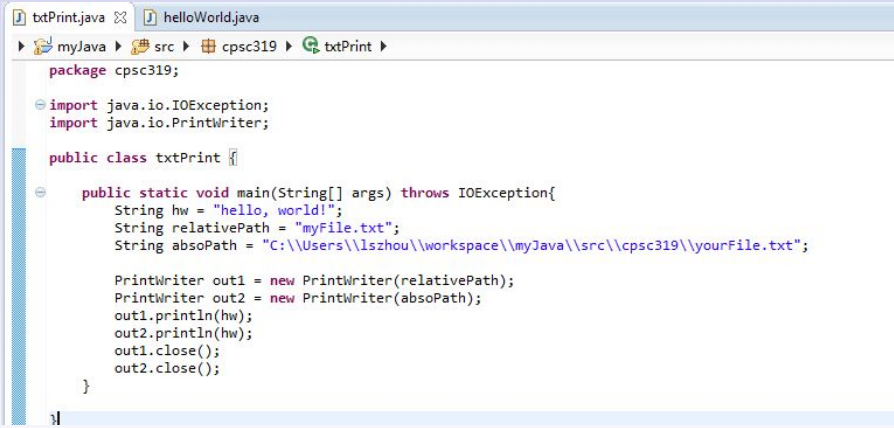
- If the file ”myFile.txt” does not exist, it is created. You can simply use the ***print, println*** and ***printf*** commands as you did when printing to System.out.

Here are some points you need to notice:

- When you specify a relative file, such as "myFile.txt", the (created) file is located relative to the directory in which the Java Virtual Machine (JVM) was started. So sometimes, to avoid confusion, absolute path is a good choice.

- If you construct a PrintWriter with a file name that can not be created, an exception occurs. ("Exception" maybe introduced later.) For now, you should simply tell the compiler that you have noticed the possibility of a "file not found" exception by tagging the main method with a throws clause, like this:

```
public static void main(String[ ] args) throws IOException
```



```java
package cpsc319;

import java.io.IOException;
import java.io.PrintWriter;

public class txtPrint {

    public static void main(String[] args) throws IOException{
        String hw = "hello, world!";
        String relativePath = "myFile.txt";
        String absoPath = "C:\\Users\\lszhou\\workspace\\myJava\\src\\cpsc319\\yourFile.txt";

        PrintWriter out1 = new PrintWriter(relativePath);
        PrintWriter out2 = new PrintWriter(absoPath);
        out1.println(hw);
        out2.println(hw);
        out1.close();
        out2.close();
    }

}
```

Figure: Output to .txt By Relative and Absolute Path

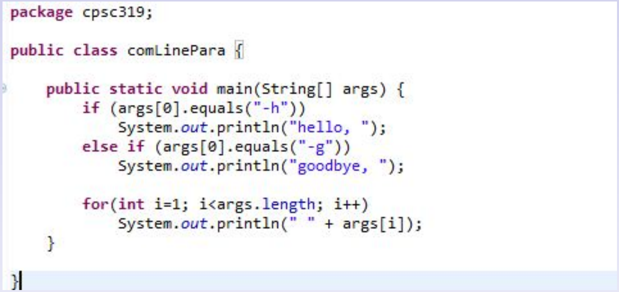
Some Open Questions for Your Thinking?

- Each time we run the above program, the content in myFile.txt will be overrided, what if we only want to add something new?
- Are there some other solutions for printing to .txt file, what are they?

- We have noticed that, generally, every java program has a *main* method with a *String[ ] args* parameters. This parameter indicates that the *main* method receives an array of strings, namely, the arguments specified on the command line.
- You will find that one requirement of your assignment is "*Your program will be invoked from the command line as follows:*"

java Assign1 order size algorithm outputfile

Let's see an example.



```
package cpsc319;

public class comLinePara {

    public static void main(String[] args) {
        if (args[0].equals("-h"))
            System.out.println("hello, ");
        else if (args[0].equals("-g"))
            System.out.println("goodbye, ");

        for(int i=1; i<args.length; i++)
            System.out.println(" " + args[i]);
    }

}
```

If the program is called as:

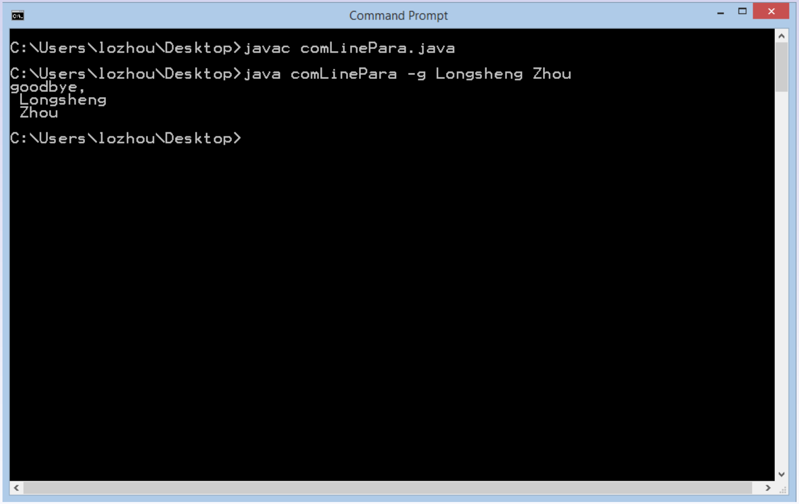
java comLinePara -g Longsheng Zhou

then the args array has the following contents:

args[0]: “-g”
args[1]: “Longsheng”
args[2]: “Zhou”

The program prints the message

goodbye, Longsheng Zhou



```
C:\Users\lozhou\Desktop>javac comLinePara.java

C:\Users\lozhou\Desktop>java comLinePara -g Longsheng Zhou
goodbye,
 Longsheng
 Zhou

C:\Users\lozhou\Desktop>
```

Thank you!

AUTHOR:	Longsheng Zhou
ADDRESS:	ICT 609e
	Department of Computer Science
	University of Calgary
EMAIL:	lozhou@ucalgary.ca