CS 1233 Object Oriented and Design

Created by Eddy C. Borera, PhD

Reminders

- Due :
    - Installing Java Compiler (8/28)
    - Hello World (8/28)
    - Chapter 1 readings (8/30)

Why Java ?

- widely used
- widely available
- object oriented language
- free

Main objective

- Develop general programming skills that are applicable to many languages.

- It is not about the language.

“ There are only two kinds of programming languages: those people always [gripe] about and those nobody uses.”– Bjarne Stroustrup

---

Topics

Topics to be covered (Zybook Chapter I):

- First Java program
- Compiling and running code
- Basic input and output
- Comments and whitespace

A. First Java program

```java
// -------------------------------
// Program prints "Hello, World!"
// Filename: HelloWorld.java
// -------------------------------

public class HelloWorld
{
    public static void main(String[] args) {
            System.out.println("Hello, World!");
    }
}
```

- class name: HelloWorld
- file name: HelloWorld.java

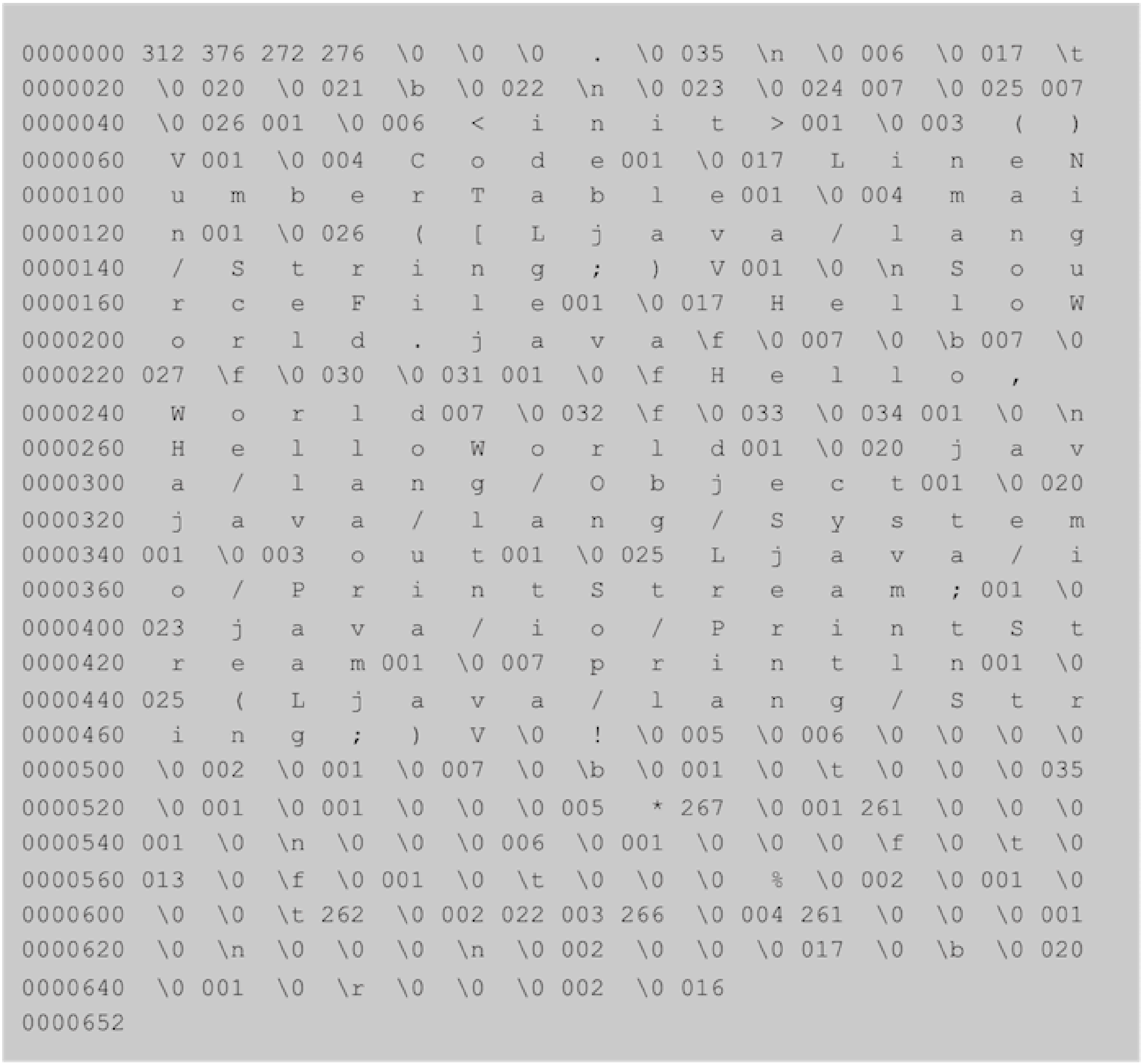
B. Compiling

- Create the program by typing it into an editor and save it as **HelloWorld.java**
- Java requires that the filename and class name to be the same
- Compile the program from a Terminal:

  ```
  $> javac HelloWorld.java
  ```

  This creates a Java bytecode file named **HelloWorld.class**

Example



```
0000000 312 376 272 276  \0  \0  \0   .  \0 035  \n  \0 006  \0 017  \t
0000020  \0 020  \0 021  \b  \0 022  \n  \0 023  \0 024 007  \0 025 007
0000040  \0 026 001  \0 006   <   i   n   i   t   > 001  \0 003   (   )
0000060   V 001  \0 004   C   o   d   e 001  \0 017   L   i   n   e   N
0000100   u   m   b   e   r   T   a   b   l   e 001  \0 004   m   a   i
0000120   n 001  \0 026   (   [   L   j   a   v   a   /   l   a   n   g
0000140   /   S   t   r   i   n   g   ;   )   V 001  \0  \n   S   o   u
0000160   r   c   e   F   i   l   e 001  \0 017   H   e   l   l   o   W
0000200   o   r   l   d   .   j   a   v   a  \f  \0 007  \0  \b 007  \0
0000220 027  \f  \0 030  \0 031 001  \0  \f   H   e   l   l   o   ,    
0000240   W   o   r   l   d 007  \0 032  \f  \0 033  \0 034 001  \0  \n
0000260   H   e   l   l   o   W   o   r   l   d 001  \0 020   j   a   v
0000300   a   /   l   a   n   g   /   O   b   j   e   c   t 001  \0 020
0000320   j   a   v   a   /   l   a   n   g   /   S   y   s   t   e   m
0000340 001  \0 003   o   u   t 001  \0 025   L   j   a   v   a   /   i
0000360   o   /   P   r   i   n   t   S   t   r   e   a   m   ; 001  \0
0000400 023   j   a   v   a   /   i   o   /   P   r   i   n   t   S   t
0000420   r   e   a   m 001  \0 007   p   r   i   n   t   l   n 001  \0
0000440 025   (   L   j   a   v   a   /   l   a   n   g   /   S   t   r
0000460   i   n   g   ;   )   V  \0   !  \0 005  \0 006  \0  \0  \0  \0
0000500  \0 002  \0 001  \0 007  \0  \b  \0 001  \0  \t  \0  \0  \0 035
0000520  \0 001  \0 001  \0  \0  \0 005   * 267  \0 001 261  \0  \0  \0
0000540 001  \0  \n  \0  \0  \0 006  \0 001  \0  \0  \0  \f  \0  \t  \0
0000560 013  \0  \f  \0 001  \0  \t  \0  \0  \0   %  \0 002  \0 001  \0
0000600  \0  \0  \t 262  \0 002 022 003 266  \0 004 261  \0  \0  \0 001
0000620  \0  \n  \0  \0  \0  \n  \0 002  \0  \0  \0 017  \0  \b  \0 020
0000640  \0 001  \0  \r  \0  \0  \0 002  \0 016
0000652
```

C. Running

- Run the bytecode from a terminal:

```
$> java HelloWorld
```

- No file extension required (i.e. without the **.class**)

D. Basic Output

Using System.out :

	Description
System.out.print("")	prints a string
System.out.println("")	prints a string followed by a new line
System.out.printf()	formatted print

D. Basic Output (1)

System.out.print() takes a string literal, which is a text between double quotes "".
Example:

```
public class KeepCalm {
   public static void main (String [] args) {

      System.out.print("Keep calm");
      System.out.print("and");
      System.out.print("carry on");
   }
}
```

Output:

```
Keep calmandcarry on
```

D. Basic Output (2)

System.out.prinlnt() takes a string literal, which is a text between double quotes "". It adds a new line at the end of the string.
Example:

```
public class KeepCalm {
   public static void main (String [] args) {

      System.out.println("Keep calm");
      System.out.println("and");
      System.out.println("carry on");
   }
}
```

Output:

```
Keep calm
and
carry on
```

D. Basic Output (3)

System.out.print() and *System.out.println()* can concatenate multiple strings before output.

```
public class KeepCalm {
   public static void main (String [] args) {

      System.out.println("Keep calm " + "and" + " carry on");
      System.out.print("Lorem " + "ipsum");
      System.out.print(" dolor sit amet");
   }
}
```

Output:

```
Keep calm and carry on
Lorem ipsum dolor sit amet
```

E. Basic Input

1. Command line arguments:
    - passed to the program before runtime
    - program arguments are seprated by space
    - arguments are stored in the "String [] args" array
    - *args[0]* contains the 1st passed argument
    - *args[1]* contains the 2nd passed argument
    - *args.length* is the number of passed arguments

E. Basic Input

Example:

```
public class Arguments {
   public static void main (String [] args) {
      System.out.println("Arguments: " + args.length);
      System.out.println("1st argument: " + args[0]);
      System.out.println("2nd argument: " + args[1]);
   }
}
```

To compile and run the program:

```
$> javac Arguments.java
$> java Arguments keep calm
```

Output:

```
Arguments: 2
1st argument: keep
2nd argument: calm
```

E. Basic Input (1)

Scanner Object

```
import java.util.Scanner
```

- it can be used to read user inputs at runtime.
- it can parse inputs from keyboard, files, string, HTTP stream, etc.
- it can parse integers, string, double, and float.

E. Basic Input (1)

```java
import java.util.Scanner;

public class BasicInput
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter name: ");
        String name = sc.nextLine();

        System.out.println("Hello " + name + "!");
    }
}
```

```
$> javac BasicInput.java
$> java BasicInput
```

```
Enter name: Eddy
Hello Eddy!
```

E. Basic Input (1)

Parsing other data types:

Types	Methods
Integer	sc.nextInt()
Double	sc.nextDouble()
String (one line)	sc.nextLine()

E. Basic Input (1)

Converting String to other types

Types	Parsing Methods
Integer	Integer.parseInt()
Double	Double.parseDouble()

```
import java.util.Scanner;

public class Parsing {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String str1 = "123";
    int num = Integer.parseInt(str1);

    String str2 = "1.05";
    double value = Double.parseDouble(str2);
  }
}
```

F. Comments and Space

- spaces/tabs between Java keywords are legal
- // comments one line
- /* multiline comments here */

```
/* ******************
 * Name: Eddy Borera
 * Class: CS 1233
 * **************** */

public class Parsing {
  public static void main(String[] args) {
    // Print welcome message
    System.out.println("Welcome to CS 1233");
  }
}
```

Escape Sequences

A character preceded by a backslash () is an escape sequence and has special meaning to the compiler.

Escape	Description
`\t`	tab
`\n`	newline
`\b`	backspace
`\'`	single quote
`\"`	double quote
`\\`	backslash

Practice (from Zybook)

Debug and fix the error:

```
public class Salary {

   public static void main (String [] args) {
      int hourlyWage = 20;

      System.out.print("Annual salary is: ");
      System.out.println(hourlyWage * 40 * 50);

      System.out.print("Monthly salary is: ");
      System.out.println((hourlyWage * 40 * 50) / 1);
      // FIXME: The above is wrong. Change the 1
      // so the statement prints monthly salary.

      return; // This is optional
   }
}
```

Practice (Canvas)

- Create a Java program named “PrintBioArguments.java” that parses four commandline arguments in the following order: first name, last name, DOB, and major.
- Example run is shown below.
- This program prints user's first name, last name, DOB,
  and major that are parsed from the command line as arguments.

  ```
  First name:        Alan
  Last name:        Turing
  DOB:                 06/23/1912
  Major:                 Mathematics
  ```

  Note: Use tabs instead of space to format the output

Extra slides

Terminal commands

	Unix	**Windows**
Navigate	`cd /path/to/folder`	`cd path\to\folder`
Navigate to parent folder	`cd ..`	`cd ..`
List file	`ls`	`dir`