

# Section 1.1

1. Write a program TenHelloWorlds.java that prints "Hello, World" ten times.

2. Describe what happens if, in HelloWorld.java, you omit

a) Error: Main method not found in class HelloWorld, please define the main method as:  
`public static void main(String[] args)`

B) Error: Main method is not static in class HelloWorld, please define the main method as: `public static void main(String[] args)`

C) Error: Main method not found in class HelloWorld, please define the main method as: `public static void main(String[] args)`

d) Error: Main method not found in class HelloWorld, please define the main method as: `public static void main(String[] args)`

3.

A) Error: Main method not found in class HelloWorld, please define the main method as: `public static void main(String[] args)`

b) same than A)

c) Exception in thread "main" java.lang.NoClassDefFoundError: vid  
at java.lang.Class.getDeclaredMethods0([Native Method](#))  
at java.lang.Class.privateGetDeclaredMethods(Unknown Source)  
at java.lang.Class.getMethod0(Unknown Source)  
at java.lang.Class.getMethod(Unknown Source)  
at sun.launcher.LauncherHelper.getMainMethod(Unknown Source)  
at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)  
Caused by: [java.lang.ClassNotFoundException](#): vid  
at java.net.URLClassLoader\$1.run(Unknown Source)  
at java.net.URLClassLoader\$1.run(Unknown Source)  
at java.security.AccessController.doPrivileged([Native Method](#))  
at java.net.URLClassLoader.findClass(Unknown Source)  
at java.lang.ClassLoader.loadClass(Unknown Source)  
at sun.misc.Launcher\$AppClassLoader.loadClass(Unknown Source)  
at java.lang.ClassLoader.loadClass(Unknown Source)  
... 6 more

d)works fine

1. Describe what happens if you try to execute [Hi.java](#) with:

(Hi, Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0

at `Hi.main(Hi.java:6)`). I cannot run the line `System.out.print(args[0]);` if I don't set first some arguments in project properties.

- a. `java Hi.`  
Works fine
  - b. `java Hi @!&^%.`  
Works fine
  - c. `java Hi 1234`  
Works fine
  - d. `java Hi.class Bob`  
Works fine
  - e. `java Hi.java Bob`  
Works fine
  - f. `java Hi Alice Bob`  
Works fine
2. Modify `UseArgument.java` to make a program `UseThree.java` that takes three names and prints out a proper sentence with the names in the reverse of the order given, so that for example, "`java UseThree Alice Bob Carol`" gives "Hi Carol, Bob, and Alice."

## Section 1.2

3. Suppose that `a` and `b` are `boolean` values. Show that the expression `(!(a && b) && (a || b)) || ((a && b) || !(a || b))` is equivalent to `true`.
4. Suppose that `a` and `b` are `int` values. Simplify the following expression: `(!(a < b) && !(a > b))`

*Solution:* `(a == b)`

5. The *exclusive or* operator `^` for `boolean` operands is defined to be `true` if they are different, `false` if they are the same. Give a truth table for this function.

`false^true= true`

`true^true=false`

`false^false=true`

6. Why does `10/3` give 3 and not 3.33333333?

For it's evaluated as integer

*Solution.* Since both 10 and 3 are integer literals, Java sees no need for type conversion and uses integer division. You should write `10.0/3.0` if you mean the numbers to be `double` literals. If you write `10/3.0` or `10.0/3`, Java does implicit conversion to get the same result.

7. What do each of the following print?

- a. `System.out.println(2 + "bc");` prints: 2bc
- b. `System.out.println(2 + 3 + "bc");` prints: 5bc
- c. `System.out.println((2+3) + "bc");` prints: 5bc
- d. `System.out.println("bc" + (2+3));` prints: bc5
- e. `System.out.println("bc" + 2 + 3);` prints: bc23

Explain each outcome.

- a) 2bc
- b) 5bc
- c) 5bc
- d) bc5
- e) bc23

Is there a trouble in the compiler? Results in b and e should be equals

8. Explain how to use [Quadratic.java](#) to find the square root of a number.  
Answer: to find the square root of c, find the roots of  $x^2 + 0x - c$ .

```
double sqroot = Math.sqrt(5);
```

9. What do each of the following print?

- a. `System.out.println('b');`
- b. `System.out.println('b' + 'c');`
- c. `System.out.println((char) ('a' + 4));`

Explain each outcome.

- a) b
- b) 197
- c) e

It's because when we try to sum several chars ('a' + 'b') the compiler takes the numeric value of it.

10. Suppose that a variable `a` is declared as `int a = 2147483647` (or equivalently, `Integer.MAX_VALUE`). What do each of the following print?

- a. `System.out.println(a);`
- b. `System.out.println(a + 1);`
- c. `System.out.println(2 - a);`
- d. `System.out.println(-2 - a);`
- e. `System.out.println(2 * a);`
- f. `System.out.println(4 * a);`

- a) 2147483647
- b) -2147483648
- c) -2147483645
- d) 2147483647
- e) -2
- f) -4

Explain each outcome.

e and f produce an overflow

11. Suppose that a variable `a` is declared as `double a = 3.14159`. What do each of the following print?

- a. `System.out.println(a);`

- b. `System.out.println(a + 1);`
  - c. `System.out.println(8 / (int) a);`
  - d. `System.out.println(8 / a);`
  - e. `System.out.println((int) (8 / a));`
- a) 3.14159
  - b) 4.14159
  - c) 2
  - d) 2.5464812403910124
  - e) 2

Explain each outcome.

If in the equation, only one of the parameters is casted, the rest of the parameters will take that type. Check C and e

**29. Day of the week.** Write a program [DayOfWeek.java](#) that takes a date as input and prints the day of the week that date falls on. Your program should take three command-line arguments: *m* (month), *d* (day), and *y* (year). For *m* use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following [formulas, for the Gregorian calendar](#):

```
y0 = y - (14 - m) / 12
x = y0 + y0/4 - y0/100 + y0/400
m0 = m + 12 * ((14 - m) / 12) - 2
d0 = (d + x + (31*m0) / 12) mod 7
```

For example, on what day of the week was August 2, 1953?

```
y = 1953 - 0 = 1953
x = 1953 + 1953/4 - 1953/100 + 1953/400 = 2426
m = 8 + 12*0 - 2 = 6
d = (2 + 2426 + (31*6) / 12) mod 7 = 2443 mod 7 = 0 (Sunday)
```

## SECTION 1.3: CONDITIONALS AND LOOPS

3. What (if anything) is wrong with each of the following statements?

```
. if (a > b) then c = 0;
```

The word **then** is not allowed in java.

- a. `if a > b { c = 0; }`
- b. `if (a > b) c = 0;`
- c. `if (a > b) c = 0 else b = 0;`

Its missing ";" after the assignation

The rest of samples are correct.

6. Suppose that `i` and `j` are both of type `int`. What is the value of `j` after each of the following statements is executed?

- a. `for (i = 0, j = 0; i < 10; i++) j += i;`

**45**

- b. `for (i = 0, j = 1; i < 10; i++) j += j;`

**1024**

- c. `for (j = 0; j < 10; j++) j += j;`

**15**

- d. `for (i = 0, j = 0; i < 10; i++) j += j++;`

**0**

8. Write a program `FivePerLine.java` that, using one `for` loop and one `if` statement, prints the integers from 1000 to 2000 with five integers per line. *Hint:* use the `%` operator.

It uses the **mod** `%` for determining when the `println` will be applied

## SECTION 1.3: WEB EXERCISES

11. What is wrong with the following code fragment?

```
double x = -32.2;
boolean isPositive = (x > 0);
if (isPositive = true) System.out.println(x + " is positive");
else                    System.out.println(x + " is not positive");
```

Just print the number and some text saying if it is positive or negative

15. What does the following program do?

```
public static void main(String[] args) {
    int N = Integer.parseInt(args[0]);
    int x = 1;
    while (N >= 1) {
        System.out.println(x);
        x = 2 * x;
        N = N / 2;
    }
}
```

Writes all powers of two greater or equal to N including 1