

CSE1325 C to Java Summary Sheet

Note: Much of the code below is in *fragments*, and will not run stand-alone.

Hello, World

```
public class HelloWorld {                // filename MUST be HelloWorld.java
    public static void main(String[] args) { // The program begins here
        System.out.println("Hello, World"); // Print to the console
    }
}
```

- The source file: HelloWorld.java (class name and filename must match *including case!*)
- To compile: `javac HelloWorld.java`
- The executable: HelloWorld.class
- To run: `java HelloWorld`

The `args` parameter is *required*. `args[0]` is the first parameter, NOT the program name. To access the command line:

```
public class HelloYou {                // filename MUST be HelloYou.java
    public static void main(String[] args) { // The program begins here
        System.out.println("Hello, " + args[0]); // Print to the console
    }
}
```

The command `java HelloYou Prof` would then print "Hello, Prof" to the console.

Primitive Types and String

primitive type	class type	common operators	example literals
int	Integer	+ - * / %	42 -17 0xFACE 0b10110010
double	Double	+ - * /%	3.14 -2.19 1.234e2
boolean	Boolean	&& !	true false
char	Character	+	'1' '&' 'π' '\n'
	String	+	"Hello" "3.14"

Class types are similar to libraries with an embedded struct. Their functions are called "methods".

```
double b = Double.MAX_VALUE;           // b = largest possible double value
double x = Double.parseDouble("1.2345"); // x = 1.2345
String s = Integer.toString(42);        // s = "42"
String q = String.valueOf(42);          // q = "42" (from the opposite side)
String c = String.format("(%d,%d)", x, y); // c = "(3,4)", like sprintf
```

Java also has a Math class (or "library"). Here's some commonly used math operations.

```
import java.util.ArrayList;
ArrayList<Boolean> TwoTruthsAndLie = new ArrayList<>(); // NOT boolean!
TwoTruthsAndLie.add(true); // append to the array list
TwoTruthsAndLie.add(false);
TwoTruthsAndLie.add(true);
boolean ttal = TwoTruthsAndLie.get(1); // set ttal to false
TwoTruthsAndLie.set(1, true); // overwrite - now they're all true
TwoTruthsAndLie.add(2, false); // insert - now true true false true
```

Input and Output

```
import System.util.Scanner;
int x = 5;
System.out.print("Hi!");           // print Hi!
System.out.println()              // print newline
System.out.println("x = " + x);    // print x = 5, followed by newline
System.out.printf("x = %d\n", x);  // print x = 5, followed by newline

Scanner scanner = new Scanner(System.in); //get a scanner
String line = scanner.nextLine();        // read the next line of text
String word = scanner.next();            // read the next word
int integer = scanner.nextInt();         // read the next integer
```

Alternately, you may use the Console.

```
Console console = System.console(); // get the system console
console.printf("Hi!");              // print Hi!
console.printf("x = %d\n", x)       // print x = 5, followed by newline

String line = console.readLine("x = %d\n y = ?", x) // print prompt, read text
int y = Integer.parseInt(line);                // convert text to int
```

You may also format to a string using `String s = String.format("x = %d", x);`

The printf / String.format codes are very similar to C, as summarized below. Square brackets denote optional parameters.

`% [flags] [width] [.precision] conversion-character`

- Flags

- - : left-justify (default is to right-justify)
- + : output a plus (+) or minus (-) sign for a numerical value
- 0 : add leading zeros (default is to use spaces)
- , : number grouping separator (for numbers > 1000)
- : space if number is positive, - if negative

- Width - minimum width of this output

- Precision - number of digits for a double, length of substring for a String

- Conversion-character

- d : decimal integer [byte, short, int, long]
- f : floating-point number [float, double]
- c : character (a capital C will uppercase the letter)
- s : String (a capital S will uppercase all the letters in the String)
- h : hashcode (a hashcode is like an address, useful for printing an object reference)
- n : newline (like n but platform-specific - Linux n, Mac r, Windows rn)

Comparisons

- Primitive values / object *addresses* are compared using == != < <= > >=
- Object *values* are compared using .equals

```
public class Test {
    public static void main(String[] args) {
        Double d1 = 3.14; // Double is class type, so this is an object
        Double d2 = 3.14; // different object, same value as d1
        if(d1 == d2) System.out.println("=="); // NOT printed!
        if(d1.equals(d2)) System.out.println(".equals"); // printed
    }
}
```

- Ternary: boolean ? if_true : if_false

```
double grade1 = 85.5;
double grade2 = 91.0;
System.out.println(grade1 > grade2 ? grade1 : grade2); // prints 91.0
```

- Conditional:

```
char letter_grade = '?';
if (grade >= 90.0) letter_grade = 'A'; // classic if / else
else if (grade >= 80.0) letter_grade = 'B';
else if (grade >= 70.0) letter_grade = 'C';
else if (grade >= 60.0) letter_grade = 'D';
else letter_grade = 'F';

switch (letter_grade) { // classic switch statement
    case 'A': // fall through
    case 'B': // fall through
    case 'C': System.out.println("Well done!"); break;
    case 'D': System.out.println("So close!\n"); break;
    case 'F': System.out.println("See you soon!\n"); break;
}

switch (letter_grade) { // new switch statement
    case 'A', 'B', 'C' -> console.printf("Well done!\n");
    case 'D' -> console.printf("So close!\n");
    default -> console.printf("See you soon!\n");
}

String result = switch(letter_grade) { // new (in-line) switch expression
    case 'A', 'B', 'C' -> "Well done!\n";
    case 'D' -> "So close!\n";
    default -> "See you soon!\n";
}
```

Loops

```
Scanner scanner = new Scanner(System.in); //get a scanner
char command = ' ';
while(command != 'x') { // while loop
    command = scanner.nextChar();
    System.out.println(command);
}

int age = 0;
do { // do loop
    age = Integer.parseInt(console.readLine("Age? "));
} while(age < 0);

for(int i = 10; i > 0; --i) // 3-term for loop
    console.printf("%d... ", i);
console.printf("\nBOOM!\n");

int[] countdown = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3};
for(int i : countdown) { // for-each loop
    if (i == 0) break; // break exits loop prematurely
    System.out.print(i + "... ");
}
System.out.println("BOOM!");
```