Lecture Week 0

Jan 5-9, 2026

Josh Hernandez

COMP 1020
Introductory Computer Science 2

University of Manitoba

Practice Problems

Hints and Expectations

- Only use Java basics from Week 0 (and your COMP 1010/1012 background).
- For “Predict the Output”, be precise about spaces, newlines, and types.
- For “Fill Code In”, change only the marked line(s) and follow constraints.
- For “Long Form”, you may write a full class with `main` or just the required method.

Practice - Predict the Output - Question 1

Integer vs double division and concatenation

```java
public class Q1 {
    public static void main(String[] args) {
        int a = 7 / 2;
        double b = 7 / 2;
        double c = 7 / 2.0;

        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("sum = " + (a + b + c));
    }
}
```

when an integer is divided by an integer, the output will be an integer

In this case, b is declared as a double, so the result will be a rounded double.

when an integer is divided by a double, the output will be a double (not rounded).

a = 3

b = 3.0

c = 3.5

3 + 3.0 + 3.5 = 9.5

What *exact* lines are printed (including decimal parts)?

Practice - Predict the Output - Question 2

String concatenation and operator precedence

```java
public class Q2 {
    public static void main(String[] args) {
        int x = 2;
        int y = 3;

        System.out.println("x + y = " + x + y); // x + y = 23
        System.out.println("x + y = " + (x + y));
        System.out.println(x + y + " = x + y");
        System.out.println("sum: " + (x + y * 2));
    }
}
```

this is a string concatenation

23 → this is "2 and 3"
→ not 2 + 3

→ numeric addition. ⇒ x + y = 5

2 + 3 = 5

5 = x + y

2 + 3 * 2
= 2 + 6 = 8

Write exactly what each of the four lines prints.

Practice - Predict the Output - Question 3

Characters, integers, and casting

```
public class Q3 {
    public static void main(String[] args) {
        char ch1 = 'A';          'A' + 2 = 'C' (char is basically a number under
        char ch2 = (char)(ch1 + 2);          the hood (Unicode)).
        int code1 = ch1;
        int code2 = ch2;

        System.out.println(ch1 + " " + ch2); // A C
        System.out.println(code1);// 65
        System.out.println(code2);// 67
        System.out.println((char)(code2 - 1)); // 'C' - 1 = B
                                   67 - 1 = 66
                                            ↳ 'B'
    }
}
```

What four lines are printed? (Remember that `'A'` has an integer code.)

Practice - Predict the Output - Question 4

Pre- and post-increment on the same variable

```
public class Q4 {
    public static void main(String[] args) {
        int n = 3;
        int a = n++; = 4 ( n now equals 4)
        int b = ++n; // never actually do this in real code!
                   = 5 (n now equals 5)
        int c = n++ + ++a; // never actually do this in real code!
                5  +  4 = 9 ( n now equals 6)
        System.out.println("n = " + n); // n = 6
        System.out.println("a = " + a); // a = 4
        System.out.println("b = " + b); // b = 5
        System.out.println("c = " + c); // c = 9
    }
}
```

- n++ (post-increment): use n first, then increase it by 1 afterward. In our expression, n contributed 5, and then n became 6.
- ++a (pre-increment): increase a by 1 first, then use it. In our expression, a became 4 first, and contributed 4.

Trace each step and write the four lines that are printed.

Practice - Predict the Output - Question 5

Short-circuit evaluation and side effects

- Case 1: A && B
  If A is false, then false && B is always false, no matter what B is. So ++y will be skipped and it doesn't increment, the y is still 1. Otherwise, when A is true, Java must evaluate B to determine the final result, so ++y will run and y will increment.

```
public class Q5 {
    public static void main(String[] args) {
        int x = 1;
        int y = 1;

        boolean a = (x++ > 1) && (++y > 1);
        boolean b = (x++ >= 2) || (++y >= 2);

        System.out.println("a = " + a); // a= false
        System.out.println("b = " + b); // b = true
        System.out.println("x = " + x); // x = 3
        System.out.println("y = " + y); // y = 1
    }
}
```

use the current value x = 1 and then increment x => after that x = 2.

"1 > 1" => false   "1 > 1" => false

=> this statement is false

=> the Statement is true

"3 >= 2" => true   " 2 >= 2 " => true

What four lines are printed? Pay attention to which increments actually run.

Practice - Predict the Output - Question 6

Nested for-loops with arithmetic

```java
public class Q6 {
    public static void main(String[] args) {
        int total = 0;

        for (int i = 1; i <= 3; i++) {
            for (int j = 0; j < i; j++) {
                total += i + j;
            }
        }

        System.out.println(total);
    }
}
```

i = 1
j = 0
total = 0 + 1 + 0
= 1

i = 2
j = 0
total = 1 + 2 + 0
= 3

i = 2
j = 1
total = 3 + 2 + 1
= 6

i = 3
j = 0
total = 6 + 3 + 0
= 9

i = 3
j = 1
total = 9 + 3 + 1
= 13

i = 3
j = 2
total = 13 + 3 + 2
= 15

What single integer is printed?

Practice - Predict the Output - Question 7

While-loop with digit processing

```java
public class Q7 {
    public static void main(String[] args) {
        int n = 4305;
        int sum = 0;
        int countEven = 0;

        while (n > 0) {
            int digit = n % 10;
            if (digit % 2 == 0) {
                countEven++;
            }
            sum += digit;
            n = n / 10;
        }

        System.out.println("sum = " + sum);
```

1st loop

digit = 4305 % 10 = 5

sum = 0 + 5 = 5

n = 4305 / 10 = 430

2nd loop

digit = 430 % 10 = 0

count Even = 1

Sum = 5 + 0 = 5

n = 430 / 10 = 43

3rd loop

digit = 43 % 10 = 3

count Even = 1

sum = 5 + 3 = 8

n = 43 / 10 = 4

4th loop

digit = 4 % 10 = 4

Count Even = 2

sum = 8 + 4 = 12

n = 0 => Stop

Practice - Predict the Output - Question 8

Array references, clone, and updates

```
public class Q8 {
    public static void main(String[] args) {
        int[] a = {1, 2, 3};
        int[] b = a; No new array is created. b and a point to the same array in memory. changing one changes what other "sees".
        int[] c = a.clone(); Create a new array with the same length and values. C and a are different arrays
                                                                         changing c does not change a.

        b[0] = 10; b = { 10, 2, 3} => a = { 10, 2, 3 }
        c[2] = 99; c = { 1, 2, 99}

        System.out.println(a[0] + ", " + a[2]); // 10, 3
        System.out.println(b[0] + ", " + b[2]); // 10, 3
        System.out.println(c[0] + ", " + c[2]); // 1, 99
    }
}
```

What three lines are printed? Explain which arrays share memory.

Practice - Predict the Output - Question 9

Modifying an array inside a static method

```java
public class Q9 {
    public static void bump(int[] data, int k) {
        for (int i = 0; i < data.length; i++) {
            data[i] += k;
        }
        k = 0;
    }

    public static void main(String[] args) {
        int[] nums = {1, 2, 3};
        int step = 5;

        bump(nums, step);

        System.out.println("step = " + step); // 5
        System.out.println(nums[0] + " " + nums[1] + " " + nums[2]);
    }
}
```

i = 0
data [0] = 1 + 5
= 6

i = 1
data [1] = 2+5
= 7

i = 2
data [2] = 3+5
= 8

// 6 7 8

What are the two lines of output? Why does one variable change and the other not?

Practice - Predict the Output - Question 10

Reasoning about a random die roll

```java
public class Q10 {
    public static void main(String[] args) {
        double x = Math.random(); // 0.0 <= x < 1.0
        int roll = (int)(x * 6) + 1;
        System.out.println(roll);
    }
}
```

The exact value is unpredictable, but:

- What is the smallest possible value that could be printed? roll = 1
- What is the largest possible value that could be printed? roll = 6

Practice - Fill in Missing Code - Question 11

Simple password rule with length and first character

A password is considered “strong enough” if:

- it has length at least 8, and
- its first character is *not* a space character.

Assume that `password` is a non-empty `String` already read from a `Scanner`.
Fill in the boolean expression so that `strong` is `true` exactly when the rules are satisfied.

```java
String password = /* already read from Scanner */;

// FILL: use only length() and charAt(0); no loops
boolean strong = ...;

System.out.println(strong);
```

Practice - Fill in Missing Code - Question 12

Two-pass array processing

Given an `int[] marks` of length at least 1, complete the code so that it prints how many marks are strictly above the average.

Fill only the lines marked ....

```
int[] marks = {72, 85, 90, 60};

int sum = 0;
for (int i = 0; i < marks.length; i++) {
    sum += marks[i];
}
                 sum / marks.length
double avg = ...;   // FILL: compute average as a double
int countAbove = 0;

for (int i = 0; i < marks.length; i++) {
    // FILL: increment countAbove only when marks[i] is strictly above
       avg If (mark[i] > avg) {
    ...       countAbove++;
}         }

System.out.println(countAbove); // should print 2 for the example array
```

Practice - Fill in Missing Code - Question 13

Scanning a String with charAt

Complete the loop so that `index` is set to the position of the first uppercase English letter (`'A'` to `'Z'`) in the string, or stays `-1` if none exists.

Fill only the condition in the `if` statement.

```java
String s = "Comp1020";
int index = -1;

for (int i = 0; i < s.length(); i++) {
    char ch = s.charAt(i);
    if (...) {  // FILL: condition for ch being uppercase A-Z
        index = i;
        break;
    }
}

System.out.println(index); // should print 0 for "Comp1020"
```

↳ `if (ch >= 'A' && ch <= 'Z')`

Practice - Fill in Missing Code - Question 14

Loop bounds and step size

Complete the loop so that it prints the sum of all even numbers from 2 to 20 (inclusive).
You must use a single `for` loop.
Fill only the for-loop header.

```java
int sum = 0;

// FILL: choose correct start, condition, and update so i takes 2, 4, ..., 20
for (...) {   for ( int i = 2 ; i <= 20 ; i += 2 )
    sum += i;
}

System.out.println(sum); // should print 110
```

Practice - Fill in Missing Code - Question 15

Overwriting negative entries in-place

You are given an `int[] data`. Move all non-negative values (`>= 0`) to the front of the same array, keeping their original order, and leave the rest of the array as-is.

Fill only the two lines marked `...`.

```java
int[] data = {3, -1, 0, 7, -2, 4};

int write = 0;
for (int read = 0; read < data.length; read++) {
    if (data[read] >= 0) {
        // FILL: copy element at index read into position write
        ... data[write] =data[read]
        // FILL: advance write index
        ...  write++;
    }
}

System.out.println(java.util.Arrays.toString(data));
// one correct result: [3, 0, 7, 4, -2, 4]
```

Practice - Fill in Missing Code - Question 16

Clamping a value into a range

Given an integer `x`, clamp it so that the result is always between 0 and 100 inclusive. Fill in the single expression using only `Math.min` and `Math.max`.

```
int x = -7;    // try changing this to test

// FILL: use Math.min and Math.max so clamped is in [0, 100]
int clamped = ...; Math.min(100, Math.max(0,x))

System.out.println(clamped);
```

Practice - Fill in Missing Code - Question 17

Combining nextLine with parseInt

You want to read a whole line that contains a student's name and age separated by a space, for example:

```
Alice 20
```

Complete the code to: (1) read the full line, (2) split it into the name and age parts using `indexOf("")`, and (3) print the name and the age plus 1.
Fill only the lines marked ....
Code on next slide.

Practice - Fill in Missing Code - Question 17 - Continued

Combining nextLine with parseInt

```java
java.util.Scanner sc = new java.util.Scanner(System.in);

String line = sc.nextLine();
int spacePos = line.indexOf(" ");

// FILL: extract the name (before the space)
String name = line. substring (0, spacePos);

// FILL: extract the age text (after the space)
String ageText = line. substring (spacePos + 1);

int age = Integer.parseInt(ageText);
System.out.println(name + " will be " + (age + 1) + " next year.");
```

Practice - Fill in Missing Code - Question 18

Aligning numbers in a table

Complete the format string so that each line printed looks like this:

```
  2 |   4 |  8.00
 10 | 100 | 32.00
```

(two columns of integers, right-aligned in width 3, and a double right-aligned in width 5 with 2 digits after the decimal).
Fill only the first argument to `printf`.
Code on next slide.

Practice - Fill in Missing Code - Question 18 - Continued

Aligning numbers in a table

```
int a = 2;
int b = 4;
double c = 8.0;
                "%3d%9d%9.2f"

System.out.printf(..., a, b, c); // FILL format string only
System.out.println();

a = 10;
b = 100;
c = 32.0;
                "%3d%9d%9.2f"

System.out.printf(..., a, b, c); // FILL format string only
System.out.println();
```

Practice - Fill in Missing Code - Question 19

Loop with early exit

Complete the loop so that `increasing` is `true` exactly when every element is strictly larger than the previous one.
Fill only the loop body.

```
int[] arr = {3, 5, 7, 10};

boolean increasing = true;
for (int i = 1; i < arr.length && increasing; i++) {
    // FILL: update increasing if a non−increasing pair is found
    ... if (arr[i] < arr[i - 1]){
                increasing = false;
}       }

System.out.println(increasing); // should print true for {3,5,7,10}
```

Practice - Fill in Missing Code - Question 20

Translating an English rule to code

You are given three integers: `hour`, `minute`, and `isWeekend` (0 for weekday, 1 for weekend).

A coffee shop is open if:

- it is a weekday and the time is between 7:00 and 18:00 inclusive, or
- it is a weekend and the time is between 8:00 and 16:00 inclusive.

Complete the boolean expression.

```java
int hour = 9;      // 0-23
int minute = 30;   // 0-59
int isWeekend = 0; // 0 = weekday, 1 = weekend

// FILL: use relational ops and &&, ||, ! only
boolean open = (isWeekend == 0 && hour >= 7 && minute >= 0 && minute <= 59 && hour <= 18 && minute >= 0 && minute <= 59) ||
               (isWeekend == 1 && hour >= 8 && minute >= 0 && minute <= 59 && hour <= 16 && minute >= 0 && minute <= 59)
System.out.println(open);
```

Practice - Write Full Code - Question 21

Weighted course grade and letter

Write Java code for a small program (or a static method) that:

- uses a `Scanner` to read three doubles: assignment average, test average, and final exam mark;
- computes a numeric course grade using weights: 40% assignments, 30% tests, 30% final exam;
- prints the numeric course grade rounded to one decimal place using `printf`;
- then prints a letter grade (A, B, C, D, F) using an `if/else if/else` chain with reasonable cutoffs that you choose.

You may assume all input is valid and between 0 and 100.

Practice - Write Full Code - Question 22

Daily step tracker

Write code that uses an `int[] steps` of length 7 (one entry per day of the week) and:

- computes and prints the total number of steps;
- computes and prints the average as a `double` with two decimal places;
- finds and prints the maximum number of steps and the index (day number) where that maximum occurred.

You can either fill the array with literals in code or read the values from a `Scanner`.

Practice - Write Full Code - Question 23

Celsius to Fahrenheit table

Write code that prints a table converting Celsius to Fahrenheit.

- Use a `for` loop where Celsius runs from -20 to 40 in steps of 5.
- For each value `c`, compute Fahrenheit as `f = c * 9.0 / 5.0 + 32`.
- Print each line using `printf` so it looks like:

```
  -5 C -> 23.0 F
```

(nicely aligned columns, to one decimal place).

Practice - Write Full Code - Question 24

Counting vowels and consonants

Write a program that:

- reads a line of text using `Scanner.nextLine()`;
- converts it to all lower-case using `toLowerCase()`;
- counts how many characters are vowels (a, e, i, o, u), how many are consonant letters (b, c, d, ..., z), and how many are "other" (spaces, digits, punctuation, etc.);
- prints the three counts.

Use only `length()`, `charAt()`, and simple comparisons; no arrays are required.

Practice - Write Full Code - Question 25

Number guessing with while-loop

Write code for a simple number guessing game with a fixed secret.

- Store a secret integer between 1 and 100 in a variable (e.g., `int secret = 42;`).
- Use a `Scanner` to repeatedly ask the user for a guess until they get it right.
- After each guess, print “too low”, “too high”, or “correct”.
- Count how many guesses were needed and print this when the user is correct.

Do *not* use `Math.random()` for this problem; the secret can be hard-coded.

Practice - Write Full Code - Question 26

Scaling an array of marks

You are given an `int[] marks` representing raw test scores where the highest mark might not be 100. Write code that:

- finds the maximum mark in the array;
- computes a scale factor so that this maximum becomes 100 (for example, if max is 40, the scale factor is 2.5);
- builds a `double[]` array of scaled marks using that factor;
- prints both the original and scaled arrays using `java.util.Arrays.toString`.

Assume all marks are positive and the array is non-empty.

Practice - Write Full Code - Question 27

Mapping integers to weekday names

Write a program that:

- reads an integer from 1 to 7;
- uses either a chain of `if/else if` or a `switch` statement to map: 1 → Monday, 2 → Tuesday, ..., 7 → Sunday;
- prints the name of the day;
- prints “invalid” if the number is not between 1 and 7.

This is a good place to practice clean, readable branching logic.

Practice - Write Full Code - Question 28

Counting grade ranges

Given an `int[] scores` of exam marks between 0 and 100 inclusive, write code that counts how many scores fall into each of these ranges: [0,49], [50,59], [60,69], [70,79], [80,89], [90,100]. Then print a small summary such as:

```
0-49:   3
50-59:  5
...
90-100:  2
```

Use only arrays, loops, and simple conditionals; no collections.

Practice - Write Full Code - Question 29

Robust user “yes/no” input

Write code that asks the user a yes/no question and reads a full line of input. Treat the answer as “yes” if, after trimming leading and trailing spaces, it equals “yes” or “y” ignoring case. Treat it as “no” if it equals “no” or “n” ignoring case. For any other input, print a message saying it was not understood.
You may use `trim()` and `equalsIgnoreCase()` on the string.

Practice - Write Full Code - Question 30

Taxi fare calculator with named constants

Design a small program that calculates a taxi fare given a distance travelled in kilometers.

- Use named constants (`final double` or `final int`) for base fare, per-km rate, and any surcharges (for example, night rate or weekend).
- Read the distance and whether it is a weekend from the user.
- Use at least one helper `static` method to compute the fare from the inputs.
- Print a nicely formatted summary including all values.

Focus on clean use of constants and methods, not on realistic pricing.