Name _____

CISC 3115 – Modern Programming Techniques
Spring 2026
Exam 0 - Solutions

Part I. Answer all questions <u>in the space to the left</u>. (60 points) These solutions are at the end of the testbank

1. Which Java keyword allows access to classes defined in another package?
a. include
b. import
c. package
d. using

2. Which of the following best describes Java program execution?
a. Direct hardware execution
b. Interpretation without compilation
c. Compilation followed by interpretation
d. Interpretation followed by linking

3. Which of the following is *not* a Java primitive type?
a. int
b. double
c. boolean
d. String

4. A constant in Java is typically declared using which keyword?
a. const
b. static
c. final
d. fixed

5. Imperative programming primarily focuses on:
a. Describing data relationships
b. Declaring object hierarchies
c. Specifying step-by-step actions
d. Defining abstract behavior

6. Which assignment operator adds a value to a variable?
a. =+
b. +=
c. ++
d. Add

7. Which construct selects one of several execution paths?
a. for
b. while
c. switch
d. break

8. Which loop guarantees execution at least once?
   a. for
   b. while
   c. foreach
   d. do-while

9. A method's signature consists of its:
   a. name and return value
   b. name and parameter list
   c. return type and body
   d. parameters and body

10. Method overloading allows methods to:
   a. Share the same name with different parameters
   b. Use different return types only
   c. Be defined in different classes
   d. Override inherited methods

11. The primary goal of method decomposition is to:
   a. Reduce memory usage
   b. Improve execution speed
   c. Break complex tasks into smaller ones
   d. Eliminate parameters

12. Which method checks for end-of-file when reading integers?
   a. eof
   b. hasNextInt
   c. nextInt
   d. isEOF

13. A prompt is primarily used to:
   a. Validate user input
   b. Display output formatting
   c. Instruct the user what to enter
   d. Flush the input buffer

14. Arrays in Java are:
   a. Dynamically resized by default
   b. Fixed-size once created
   c. Passed by value
   d. Primitive types

15. Which syntax creates an integer array of size 10?
   a. int arr = new int(10);
   b. int arr[10];
   c. int[] arr = new int[10];
   d. array int arr = 10;

16. Which method returns the number of characters in a String?
a. size
b. count
c. length
d. charAt

17. Which method finds a substring inside another string?
a. find
b. indexOf
c. search
d. contains

18. Which technique determines whether one number divides another evenly?
a. Modulo
b. Casting
c. Swapping
d. Sorting

19. Swapping two values typically requires:
a. One variable
b. No temporary storage
c. A temporary variable
d. Only arithmetic

20. Parallel arrays are used to:
a. Increase capacity
b. Store related data in separate arrays
c. Eliminate sorting
d. Replace objects

21. A sentinel approach is discouraged because it:
a. Uses extra memory
b. Requires sorting
c. Assumes impossible input values
d. Is slower

22. EOF-based input termination relies on:
a. A special marker value
b. File metadata
c. Scanner state
d. User confirmation

23. Which error occurs when dividing by zero?
a. Syntax error
b. Logic error
c. Compilation error
d. Runtime error

24. After sorting, why can occurrences of the same value be counted with a single scan
   a. The values are evenly distributed
   b. Equal values appear next to each other
   c. Sorting removes duplicates
   d. The array becomes smaller

25. Which statement about arrays is true?
   a. They resize automatically
   b. They store mixed types
   c. Their length cannot change
   d. They require Scanner

26. Which Java feature groups related code and data?
   a. Methods
   b. Classes
   c. Packages
   d. Arrays

27. Which loop is best when the number of iterations is known?
   a. while
   b. do-while
   c. for
   d. foreach

28. Which value type is returned by relational operators?
   a. int
   b. char
   c. boolean
   d. double

29. Which array access is guaranteed to cause an ArrayIndexOutOfBoundsException?
   a. arr[length]
   b. arr[0]
   c. arr[1]
   d. arr[length-1]

30. Which Java feature allows multiple methods with the same name?
   a. Inheritance
   b. Polymorphism
   c. Overriding
   d. Overloading

Part II. Answer all questions (40 points).

31. **(5 points)** Given the method
`boolean allAreEqual(int a, int b, int c)` that returns true if all three values are equal, write the method
`boolean twoAreEqual(int a, int b, int c)`
that returns true when exactly two of the numbers are equal. For full credit, you must use the `allAreEqual` method.

This method is straight out of Lab 0.1. I am also presenting the allAreEqual method:

```
public static boolean allAreEqual(int n1, int n2, int n3) {
        return n1 == n2 && n2 == n3;
}

public static boolean twoAreEqual(int n1, int n2, int n3) {
        return !allAreEqual(n1, n2, n3) &&
                (n1 == n2 || n1 == n3 || n2 == n3);
}
```

1 point deducted if you didn't leverage the method

32. **(5 points)** Write the method
`boolean noneAreEqual(int a, int b, int c)`
that returns true when none of the values are equal. For full credit you must use both the `allAreEqual` and the `twoAreEqual` methods from the previous question

Also straight out of 0.1

```
public static boolean noneAreEqual(int n1, int n2, int n3) {
        return !allAreEqual(n1, n2, n3) && !twoAreEqual(n1, n2, n3);
}
```

1 point deducted if you didn't leverage the methods

33. (5 points) Given a `Scanner` variable `sc`, write some code that reads header values until eof. Each header value is followed by that number of integers. Print out the numbers (not the header value) and their sum. You may not assume the header value and the numbers that follow are on the same line. For example, if the file contains:

```
3 5 10 20
2 9
   1
```

The output would be:

```
      5 10 20 : 35
      9 1 : 10
```

This is Lab 0.2, but asking for the sum, rather than the average, of the sequence

```
int count = 0;
while (scanner.hasNextInt()) {
   int howMany = scanner.nextInt();
   int total = 0;
   for (int i = 0; i < howMany; i++) {
      int number = scanner.nextInt();
      total += number;
      System.out.print(number + " ");
   }

   System.out.println("… the sum is is " + total);
}
```

34. (5 points) Write a method `int find(Scanner sc, String s)`. The method reads string values from the scanner, and returns the position (starting at 1) of the value of `s` if it is found, and -1 otherwise. For example, if the file corresponding to the scanner contains:

`"Hello" "Goodbye" "Wecome" "Farewell"`

and `s` contains the value "`Welcome`", the method returns 3.

Basically Lab 0.3, but here I am asking for the first occurrence (easier somewhat than last occurrence). No points deducted if you did last occurrence. Most common mistake was == instead of `.equals`. Another common issue was using `nextLine` improperly.

```
public static int findLast(Scanner sc, String s) throws Exception {
   int count = 0;
   while (sc.hasNext()) {
      String inputStr = sc.next();
      count++;
      if (inputStr.equals(s)) return count;
      count++;
   }
   return -1;
 }
```

(20 points) Write a full app that reads values from the file "`data.text`" and loads them into an array (you may assume no more than 100 values). The app should then:

- print the number of values read
- print the number of positive and negative numbers are in the array
- print the largest number in the array
- print the array

You are free to write your app in any way you please, but I would be happy to see at least a `read` method (to read the data into your array) and a `print` method (to print the array).

Essentially Lab 0.4 (I am leaving this blank so you can still try to do it if you had trouble. Most common mistake: assuming the array was fully rather than partially populated and thus used a.length instead of actual size (number of values read in) t process the array.