1. Lab solution Average is 10% of overall grade (10 at 1% each)

   Lab Report Average is 10% of overall grade (10 at 1% each)

   Homework Average is 80% of overall grade (8 at 10% each)

   Final exam is optional and replaces a lower, previous homework assignment (10%*)

2. Blackboard

3. Policies

Late work: **We do not accept late work in this course**. Assignments are made available to everyone at the same time and are due at the same time. No credit will be given for late assignments. Exceptions to the late policy may be made on an emergency basis. Furthermore, all absences due to **medical**, **family**, or **other significant emergencies** must be excused by the Ombudsman's office. All documentation, including but not limited to doctor's notes, obituaries, etc., must be sent to them and approved before any exception can be made.

Make up work: Make-up work for assignments can be requested before the assignment is due. If there is a known conflict that will prevent submitting work at the assigned time, then the instructor must be informed **before the deadline**. These requests must be done electronically via email. Only a small fraction of assignments can be made up for any excuse(s), which is limited to exactly **1 Homework, 1 Lab Solution, and 1 Lab Report.** Outstanding make-up work must be completed and submitted before any additional make-up work can be issued.

Regrade requests may only be made within **ONE WEEK** after the assignment has been graded. These requests must be done *electronically* via email to the instructor. Regrade requests only apply to previously submitted work and we do not accept additional work after the fact. Multiple, frivolous regrade requests will disqualify future regrade requests.

4. True

5. All programming assignments (labs solutions, homework, and exam solutions) are to be written in the Java programming language and require the source files (JAVA file extension) to be submitted (.java)

6. Byte code

7. Syntax error
   Java does not allow implicit casting from a double (floating-point type) to an int (integer type) because it involves a loss of precision.
   You must explicitly cast double to int to indicate that you are aware of the potential data loss.
   Logic error: Even if you fix the syntax error by casting, the fractional part (.8) will be truncated (not rounded). Thus, j = 1024.8 will become 1024 when assigned to i.

   To fix: explicitly cast j to int

   ```
   double j = 1024.8;
   int i = (int) j; // Explicit cast
   System.out.println(i);
   ```

   Output after fix: 1024

8. **Syntax Error**:

   The line int[] array = new array[10]; is incorrect.

   array is used as a type, which is invalid.

   It should be corrected to int[] array = new int[10];.

   Run-Time Error:

   The second for loop condition i <= array.length causes an ArrayIndexOutOfBoundsException.

   array.length gives the size of the array (10), but valid indices are 0 to 9.

   The condition should be i < array.length instead of i <= array.length.

   Logic Error: fixing the above errors, the code will work as intended and print the values stored in the array. There are no logic errors after addressing the other issues.

9. No syntax errors, but there is logic errors
   Logic Error:

- **Issue with the while loop condition**:
  - The semicolon (;) after the while statement makes it an **empty loop**.
  - This means the block of code is **not part of the while loop**. Instead, it is treated as a separate block that executes unconditionally after the while loop.
- **Issue with the while loop condition (j < 0.0)**:
  - The condition j < 0.0 will **never be true** because j is initialized to 10.0 and is not modified inside the empty loop.
  - As a result, the while loop does nothing, and the block { ... } will execute once, printing 10.0 and decrementing j.

If the intent is to decrement j and print its values while j is greater than or equal to 0.0, the code should be written as follows:

java

CopyEdit

```
double j = 10.0;

while (j >= 0.0) {

  System.out.println(j);

  j -= 0.01;

}
```

1. Remove the semicolon (;) after the while statement.
2. Adjust the loop condition as necessary (e.g., while (j >= 0.0) for the intended behavior).

Output of the Corrected Code:

The corrected code will print the values of j from 10.0 down to 0.0 in decrements of 0.01.

It would output 10.0

10. This is error free, It would output :

```
B
D
```