

# Extra Credit Assignment

## WORKING WITH JAVA PACKAGES

### VERSION 1.0

The objective of this assignment is to get you comfortable with packages and their related file structures. The programming assignment should be implemented in Java. You are required to work alone on this assignment. This assignment will contribute a maximum of 3% of extra credit towards your final grade.

This assignment may be modified to clarify any questions (and the version number incremented), but the crux of the assignment and the distribution of points will not change. If there any changes to the assignment, all changes will be document in the "Change History" section of this assignment.

**DUE DATE:** Wednesday, April 24<sup>th</sup> @ 8:00 pm

## 1 Description of Task

As part of this assignment, you will be responsible for performing the following tasks in separate Java packages. The programming assignment should be developed in Java, and your classes must reside in the package **cs250.ec**.

### Task 1

The first task involves converting a decimal number between 1 and 0 to binary. You are converting the number from base 10 to base 2. **You are not being asked to derive the XIEEE-754 floating point representation.**

The range of numbers your program should be able to handle is  $(0, 1)$  exclusive. That is, you will not be tested with numbers greater than 1 nor negative, nor 0. You can assume that you will always be given a decimal number beginning with "0."

You may encounter floating point error. An example of this is expecting a number like "0.6" and getting number like "0.6000000000000001". To handle this case, you can multiply the number by a large integer, then divide it by the same number as a double. For example,

```
value = Math.round(value * 1000000) / 1000000.0;
```

### Built in java functions that you should not use for this task:

```
Double.parseDouble(String num)
```

```
Float.parseFloat(String num)
```

Anything with similar functionality

**Command line execution:** `java cs250.ec.convert.DecimalToBinary <decimal-number>`

## Task 2

The second task involves creating a histogram of letters. Given a sentence as input count the number of occurrences of each letter. The letters are case insensitive meaning an occurrence of "A" and an occurrence of "a" should result in 2 occurrences of "a". The sentence will only contain whitespaces and letters a-z and A-Z. There will be no numbers, punctuation marks, etc.

The input sentence must be enclosed in double quotes in the command line. For example, "I am taking Computer Foundations" is valid input but not I am taking Computer Foundations.

You are required to print the occurrences in descending order. If more than one letter has the same count, print those letters in alphabetical order. You are not allowed to use an existing sorting utility in Java (e.g. Collections.sort), you must implement a sorting algorithm (e.g. bubble sort).

**Command line execution:** `java cs250.ec.counting.AlphaHistogram "<sentence>"`

## Task 3

The third task involves implementing the following equation:

$$f(x, y) = \frac{\log_{10}(|xy|) - x^4}{\sqrt{(xy)^2 + y^3 \cdot x}}$$

Both  $x$  and  $y$  will be integers. The result will be a decimal printed to 5 decimal places. You will need to implement the function using the correct order of operations. You are allowed to use the Java Math library.

**Command line execution:** `java cs250.ec.order.Equation <x> <y>`

## 2 Example Outputs

**Command:** java cs250.ec.convert.DecimalToBinary 0.5

0.5 -> 0.1

**Command:** java cs250.ec.convert.DecimalToBinary 0.625

0.625 -> 0.101

**Command:** java cs250.ec.counting.AlphaHistogram "I am taking Computer Foundations"

a: 3  
i: 3  
n: 3  
o: 3  
t: 3  
m: 2  
u: 2  
c: 1  
d: 1  
e: 1  
f: 1  
g: 1  
k: 1  
p: 1  
r: 1  
s: 1

**Command:** java cs250.ec.order.Equation -2 -5

$f(x,y) = -0.05769$

**Command:** java cs250.ec.order.Equation 70 5

$f(x,y) = -2638.46126$

### 3 What to Submit

Use the CS250 *Canvas* to submit a single .zip file that contains:

- Your java source codes, matching this directory structure:
  - cs250
    - ec
      - convert
        - DecimalToBinary.java
      - counting
        - AlphaHistogram.java
      - order
        - Equation.java
- A README.txt file containing a description of each file and any information you feel the TAs need to grade your program.

**Filename Convention:** The archive file should be named as <FirstName>-<LastName>-EC.zip . E.g., if you are Cameron Doe then the zip file should be named Cameron-Doe-EC.zip.

### 4 Grading

The assignments must compile and function correctly on machines in the CSB-120 Lab. Assignments that work on your laptop on your particular flavor of Linux, but not on the Lab machines are considered unacceptable.

This assignment will contribute a maximum of 3 points of extra credit towards your final grade. **3 points** for correctly performing Tasks 1, 2, and 3. To score at least a point, you need to have implemented at least two tasks correctly. **There will be no partial credit for implementing only one task.**

In order to receive credit, you must submit both to the test server and to Canvas.

Note that the formatting for each task should match the example outputs exactly.

You are required to **work alone** on this assignment.

## 5 Late Policy

There isn't a late submission window for this extra credit assignment. We will not be accepting any submissions past the due date. If you have university SDC accommodations, the due date will be adjusted accordingly in Canvas.

## 6 Version Change History

This section will reflect the change history for the assignment. It will list the version number, the date it was released, and the changes that were made to the preceding version. Changes to the first public release are made to clarify the assignment; the spirit or the crux of the assignment will not change.

Version	Date	Comments
1.0	4/6/24	First public release of the assignment.