Oscillations and Waves Exam 3

April 29, 2025

This exam tests your fluency with the core of the Wolfram Language, as it was presented in *An Elementary Introduction to the Wolfram Language, 3rd Edition (EIWL3),* Sections 25-34 and 38-41. There is one problem with two or three parts corresponding to each section. ***Tip: all of them are meant to be quick. If you get bogged down, move on.***

Directions:

After downloading this notebook, rename it with your first name in the filename. E.g., Eli-Exam3.nb, Harper-Exam3.nb, Hexi-Exam3.nb, Jeremy-Exam3.nb, Rania-Exam3.nb, Tahm-Exam3.nb, or Walker-Exam3.nb.

Then disconnect from the wifi and work the exam. Save your notebook early and often so that you don't lose work in progress.

Your answers always go into the Wolfram Language Input cells that begin with a comment, e.g.,

```
(* 1a *) foobar /@ Plus[Array]
```

All your answers should execute and re-execute without warnings or error messages.

You may refer to your downloaded copies of EIWL3, and anything else we developed in the course (like your cheat sheets!), but not to any web resources.

When you are done, save your notebook one last time, re-join the wifi, and then email it to me.

This exam was designed to require about 45 minutes, but if you need a full hour, that is ok. Everyone will stop at the one-hour mark.

1. Applying Functions (*EIWL3* Section 25)

(a)

Use **`Map`** with a *levelspec* to put a frame around each individual number in the array **`Array[Plus,{10,10}]`** (we don't want frames around already-framed things — just one level of frames around the individual numbers).

```
(* 1a *)
```

(b)

Copy what you did in (a), but for this part, also turn the result into a grid using **Grid** and the "as an afterthought" syntax:

```
(* 1b *)
```

2. Pure Anonymous Functions (*EIWL3* Section 26)

(a)

Use the **#** and **&** notation to create an anonymous function that cubes whatever is given it, and then use **/@** to apply it to every member of the list **{1,2,3,4,5}**.

```
(* 2a *)
```

(b)

Use the **#1**, **#2**, and **&** notation to create an anonymous function that divides its first argument by its second argument. Combine this with **Apply** and a *levelspec* to apply the function to **{{1,2},{2,3},{3,4},{4,5}}**. Once you have this right, you will get $\left\{\frac{1}{2},\frac{2}{3},\frac{3}{4},\frac{4}{5}\right\}$.

```
(* 2b *)
```

3. Applying Functions Repeatedly (*EIWL3* Section 27)

(a)

Use **Nest** to apply **Factorial** twice to **{1,2,3,4}**. If you have this right, 620,448,401,733,239,439,360,000 will be one of the elements of your answer.

```
(* 3a *)
```

(b)

Use **NestList** to apply **Factorial** three times to **{1,2,3}**, as well as showing the results of doing it 0, 1, and 2 times. If you have this right, you will have an insanely large result at the third step. Do not go any higher, or I do not know what will happen to your computer.

```
(* 3b *)
```

4. Tests and Conditionals (*EIWL3* Section 28)

(a)

Use **PrimeQ** and **/@** to generate a **True** or **False** list that is twenty elements long expressing which numbers in **Range[20]** are prime.

```
(* 4a *)
```

(b)

Combine **PrimeQ** with **Select** to only list the numbers in **Range[20]** that are prime.

```
(* 4b *)
```

5. More About Pure Functions (*EIWL3* Section 29)

(a)

Accomplish exactly the same thing as **Table[n*(n-1)/2,{n,6}]** using **Array** and a pure function.

```
(* 5a *)
```

(b)

Make some modifications to **FoldList[Plus,{1,2,3,4,5}]** so that it produces a list of the first 10 factorials. Instead of hand-coding the list up to 10, begin by first changing **{1,2,3,4,5}** to **Range[10]**.

```
(* 5b *)
```

6. Rearranging Lists (*EIWL3* Section 30)

(a)

Use **Transpose** and one of the *levelspec* options to turn
{{{1,uno},{2,dos},{3,tres}},{{4,cuatro},{5,cinco},{6,seis}}} into
{{{1,2,3},{uno,dos,tres}},{{4,5,6},{cuatro,cinco,seis}}}

```
(* 6a *)
```

(b)

Use **Flatten** and a *levelspec* option to turn
{{{1,uno},{2,dos},{3,tres}},{{4,cuatro},{5,cinco},{6,seis}}} into
{{1,uno},{2,dos},{3,tres},{4,cuatro},{5,cinco},{6,seis}}

```
(* 6b *)
```

7. Parts of Lists (*EIWL3* Section 31)

(a)

Use the magical **All** position (you will need to use **All** more than once) to turn
{{{Eli, Lerner},{Harper,Yonago},{Hexi,Jin}},{{Jeremy,Choy},{Rania,Zaki},{Tahm,Loyd},{Walker,Harris}}} into
{{Eli,Harper,Hexi},{Jeremy,Rania,Tahm,Walker}}

```
(* 7a *) foo[[3]]
```

(b)

Use a magical *negative positional argument* to extract **{Jeremy,Rania,Tahm,Walker}** from **{{Eli,Harper,Hexi},{Jeremy,Rania,Tahm,Walker}}** and combine that with **Take** with a different magical *negative* argument to extract **{Tahm,Walker}**.

```
(* 7b *)
```

8. Patterns (*EIWL3* Section 32)

(a)

Use **Cases** to choose the lists that begin and end with the same letter in this list of lists (but look ahead to part (b) before you solve part (a)):

```
{
 {"a", "l", "u", "l", "a"},
 {"a", "l", "o", "h", "a"},
 {"a", "r", "a", "r", "a"},
 {"b", "o", "n", "u", "s"},
 {"c", "i", "v", "i", "c"},
 {"d", "e", "b", "e", "d"},
 {"e", "l", "b", "o", "w"},
 {"z", "a"},
 {"z", "z"}
}
```

```
(* 8a *)
```

(b)

The pattern **BlankNullSequence** has the shorthand **___**. Use **___** to improve the pattern you used in Part (a) so that the two-letter list **{z,z}** is also included in your result.

```
(* 8b *)
```

9. Assigning Names to Things (*EIWL3* Section 38)

(a)

Use **Module** to compute **x=Factorial[10]**, and then produce **{x,x^2,x^3}**.

```
(* 9a *)
```

(b)

Inside **Module**, let **rangeSquared=Range[10]^2**, and then produce a list line plot of **rangeSquared** joined with **Reverse[rangeSquared]**.

```
(* 9b *)
```

10. Immediate and Delayed Values (*EIWL3* Section 39)

(a)

Make a *one-character change* to this expression,
Module$\left[\{x:=\text{RandomInteger}[10]\},\left\{x,x^2,x^3,x^4\right\}\right]$, so that it produces four different powers of the same random number instead of four different powers of different random numbers.

```
(* 10a *)
```

(b)

Make a *one-character change* to this expression,
Module[{color=RandomColor[]},Graphics[Table[Style[Disk[{i,0}],color],{i,5}]]], so that it produces five different-color disks.

```
(* 10b *)
```

11. Defining Your Own Functions (*EIWL3* Section 40)

(a) Wording on 11(a) could use improvement.

Define a function **f** that takes a list of three elements and out of them makes a list of lists that contains all six possible orderings. Using **Permutations** will make this easy.

Include a test of your function as **f[1,2,3]** and make sure it gets
{{1,2,3},{1,3,2},{2,1,3},{2,3,1},{3,1,2},{3,2,1}}.

```
(* 11a *)
```

(b) Wording on 11(b) could also use improvement.

Define a function **g** that gives **1** for **g[0]**, and gives **n*g[n-1]** for any integer **n** greater than **0**, *but don't use an* **If** *statement!* Include a test of your function as **g[6]** and make sure it gets **720**.

```
(* 11b *)
```

12. More About Patterns (*EIWL3* Section 41)

(a)

Use the replacement rule notation — e.g., **/.** and **->** — to exchange the first and last element in any list containing two or more elements and test your replacement using the list **{alpha, beta, gamma, delta, epsilon}**.

```
(* 12a *)
```

(b)

Starting with **Characters/@RomanNumeral[Range[100]]**, select all the sequences corresponding to the Roman numerals that have XXX in them.

```
(* 12b *)
```

(c)

Use **StringJoin** to turn what you got in 12(b) into **{XXX,XXXI,XXXII,XXXIII,XXXIV,XXXV,XXXVI,XXXVII,XXXVIII,XXXIX,LXXX,LXXXI,LXXXII,LXXXIII,LXXXIV,LXXXV,LXXXVI,LXXXVII,LXXXVIII,LXXXIX}**.

```
(* 12c *)
```