**Rust Programming Lab #7 20th September 2022**

**Revision Lab**

Since a mid-term test will soon hit you, there will be no new syntax, semantics or ‘features’ in this lab: it will ask you to write *many small fragments* of code and ensure that you understand them well enough to make the compiler accept your code. Generally, you should write the suggested code, compile it, observe any error(s) and make sure you know how to fix them! For exampe

**Exercise 1:** **mut** keyword …. Add to your ‘program’

**let n = 2; n = n + 5;**

Observe the compiler error, check that you know *how* to correct it and then correct it. At the same time, check that you *understand* what Rust requires, so that you can answer a (possible) exam question about this!

If you don’t understand your ‘fix’, ***do not simply hack a solution,*** discuss with your classmates or ask a TA to help, a correct answer may be needed in the exam 😊.

Many of the exercises suggest using random numbers.

**let x:f64 = rand::random(); // generates a number in (0,1)**

Don’t forget to add the **rand** crate to your manifest:

**rand = "0.8.5"**

**Exercise 2:** Arithemetic: Try this

**let m = 10; let x:f64 = 1.0; let x10 = x\*m;**

If it won’t compile, fix it!

**Exercise 3:** Declare an array of **N** integers.

**Exercise 3a:** The capital **N** was a strong hint that you should add ……?

**Exercise 4:** Write a loop that ***captures*** **M** random floats.

**Exercise 5:** Write a function that computes the mean of thefloats in the array from Ex 4 and returns the mean.

**Exercise 6:** Add a loop to count occurences of **N** <anything>. Try setting **N** to 4 and find the number of random numbers (from Ex 4) in each quartile.

**Exercise 7:** Compiler complains about this – why?

**let m = 20.0; let x:f64 = 1.0; { let x10 = x/m; }  
println!(“{}% is {}”, 1/m, x10);**

**Exercise 8:** Go back to Ex 2 – why does Ex 7 not give the same error?

**Exercise 9:** Tell your program to print out the maximum and minimum values for an **f32** and an **f64** (Simple way, without looking up the ranges in the documentation, but looking up the needed function or constant name(s) is OK 😊)

**Exercise 10:** Define an ‘**Optimism**’ struct that records (a) your name, (b) expected salary when you graduate and (c) a probability that you will achieve that. (Optimistic numbers are OK 😊).

**Exercise 10a:** Initialize this struct.

**let opt = …..**

**Exercise 10b:** As alternative to 10a, ***add*** *a function* that creates this struct and returns it.

**Exercise 11:** Make a structure to hold a global (GPS) position, e.g. 13.7506° N, 100.7943° E.

**Exercise 11a:** Add a constant to hold the Unicode degree symbol °, 00B016.

**Exercise 11b:** Make a function to print a GPS position, including the degree symbol and direction (N, S, E, W). Note a negative longtitude represents W from Greenwich and should be printed as positive + ‘W’, similarly, negative latitude is S.

**Exercise 12:** Define a vector of GPS points. Use the structure you defined in Ex 11.

**Exercise 12a:** Now fill this vector with a grid of GPS points. Add a constant to define the grid, e.g. number of points N or E (not too large – 4 will do 😊) and the stepsize (*e.g.* 0.01° = ~18.5m in latitude). Set some starting point for this grid, e.g. our local position – we are going to measure the spread of the flood 😊.

**Exercise 12b:** Print out the 2D matrix of grid points.

**Exercise 12c:** Change the previous loop **to remove** the points as they were used.

**Exercise 13:** Using the **Point** struct already used, design a struct for a **Polygon.** How general can you make this?

**Exercise 14:** Add a function to compute and return the perimeter of a **Polygon. Hint:** don’t forget you can (*or* ***should*)** use additional functions to simplify this task**.** Note Rust will probably insist that you pass a reference, &Polygon, to your perimeter function.

**Exercise 15:** You can add your name to your program with

**let name = “<your name – Latin or** ภาษาไทย **or中文 or …>”;**

**Exercise 16:** Make a structure that includes your name as a **string** and your age. Initialize that structure using the name you entered in Ex 15

**Exercise 17:** Make a function that takes a slice of the name string and reports the position of some character, e.g. a comma ‘,’, in it. It should return -1, if the character is not found.

**Exercise 18:** Show that you know how to direct output to the **stdout** OR **stderr** streams.

**Exercise 19:** What type would you use to label the state of the KMITL campus, e.g. flooded, wet, moist, humid, dry, *etc.*?

**Tick off the exercise on the lab sheet (✓ or 🗶 =** *if didn’t get it***) or answer the question asked, ask a TA to sign it off and hand it in.**

**Website: kris.kmitl.ac.th/clinic/Courses/Rust/**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendance** | **01286120** | **Elementary Systems Programming** | **20 Sep 2022** |

|  |  |  |
| --- | --- | --- |
| **Name (Thai script\*)** |  | **Student ID** |
| **(Latin characters -  as you enrolled)** |  |
| **\****Please write clearly: practice for one farang who is trying to improve* **😉** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Exercise | Comment | Completed  (Your check) | **TA** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3a, b | What did you add? |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10a,b,c | struct … |  |  |
| 11a,b | struct ..  Other functions used (specification only) |  |  |
| 12a,b,c | 12c: what did you change to? |  |  |
| 13 | struct …… |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 | Did you test your function? Was it correct? |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
|  |  |  |  |
|  |  |  |  |