Experiment Number: 03

Aim: Tutorial on introduction to Haskell programming environment

Lab Objective: Design and implement declarative programs in functional and logic programming languages

Lab Outcome Mapped: Design and Develop solution based on declarative programming paradigm using functional and logic programming (LO2)

Requirements: Any text editor to be able to edit Haskell code and Glasgow Haskell Compiler 8.0+ version.

Theory:

Functional Programming: Functional programming is a programming paradigm — a style of building the structure and elements of computer programs — that treats computation as the evaluation of mathematical functions and avoids changing-state and mutating data.

Advantages of functional programming include

- Functional programming is based on mathematical functions
- Easier to determine inputs
- Easier to determine outputs
- Easier to demonstrate prove that you have a correct program
- Easier to test programs that are too difficult to prove

Limitations of Functional Programming include-

- Functional programming paradigm is very different from imperative paradigm, so it is difficult to understand for the beginner.
- Hard to maintain as many objects evolve during the coding.
- Needs lots of mocking and extensive environmental setup.
- Re-use is very complicated and needs constantly refactoring.
- Objects may not represent the problem correctly

Haskell is a purely Fuctional programming language.

Haskell Features-

- PURE Functions: Functions are pure in Functional Programming.
- A function called multiple times with the same arguments will always return the same value.

- Functions are first class memebrs and can be higher order
- It allows you to handle functions as if they were normal data types.
- Functions can either accept another function as argument or can return a function themselves.
- Variables are Immutable

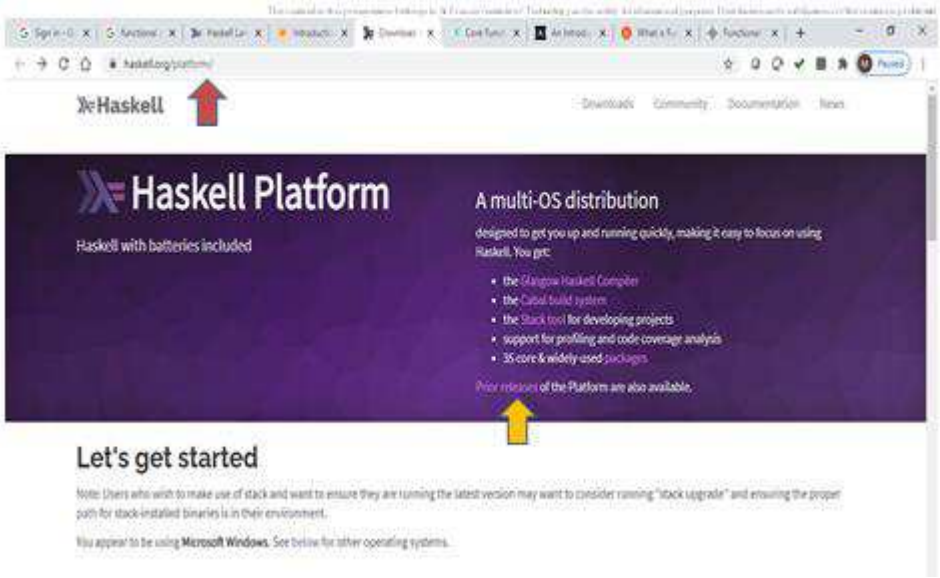
Performance steps:

As part of this tutorial students have to undertake following steps:

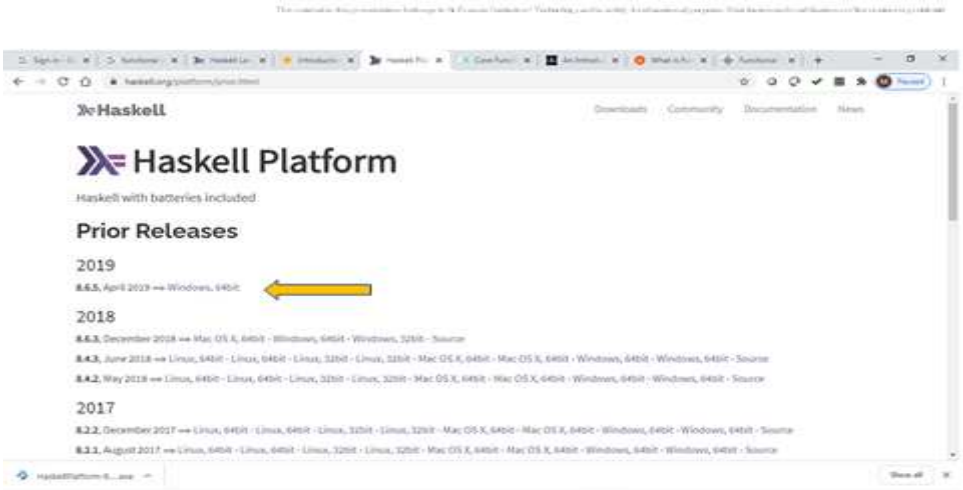
1. Download and install Haskell Compiler
2. Perform basic arithmetic operations
3. Perform basic logical operation
4. Perform basic comparison operation

Part 1: Procedure for installation of Haskell compiler-

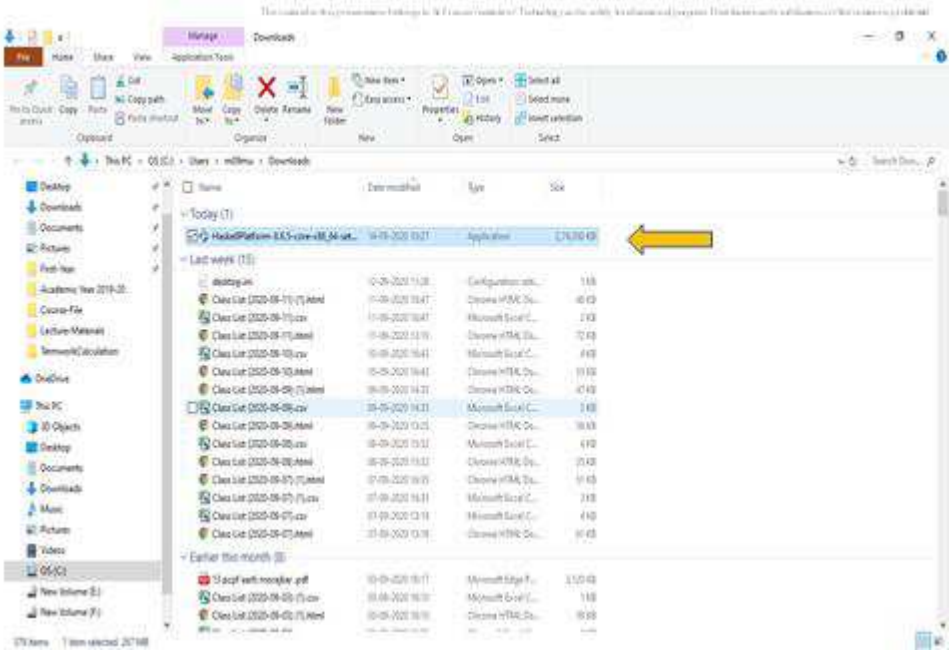
`Step-1: Visit https://www.haskell.org/ and click on prior releases



Step-2: Download 8.6.5 (April 2019) release for 64 bit windows machine

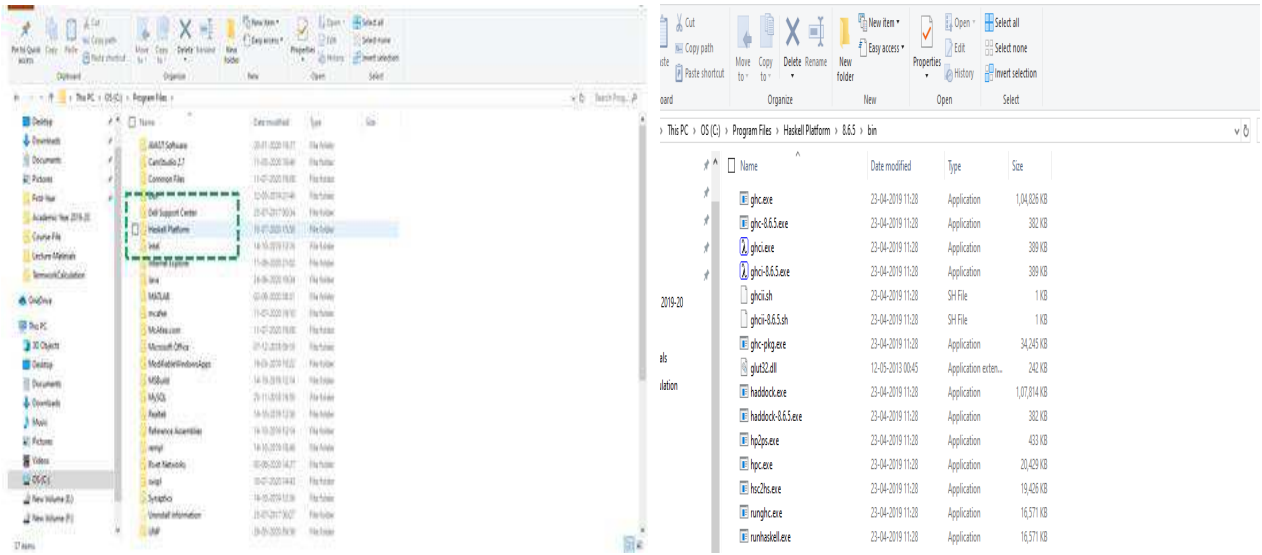


Step-3: In the Downloads folder, run the application as an administrator



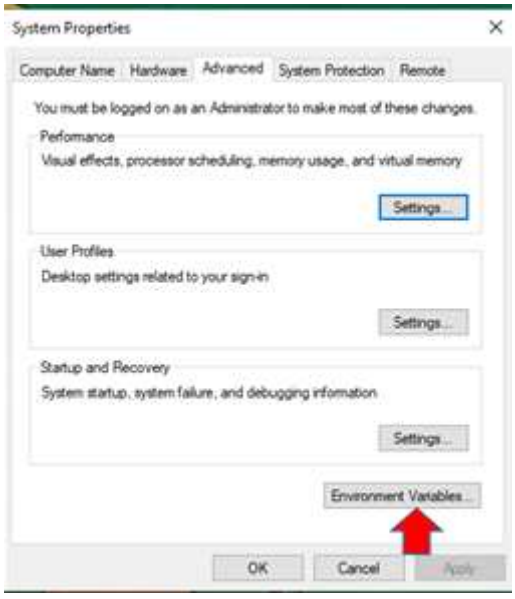
After successful installation it will be loaded in the path ‘C:\Program Files\Haskell Platform’

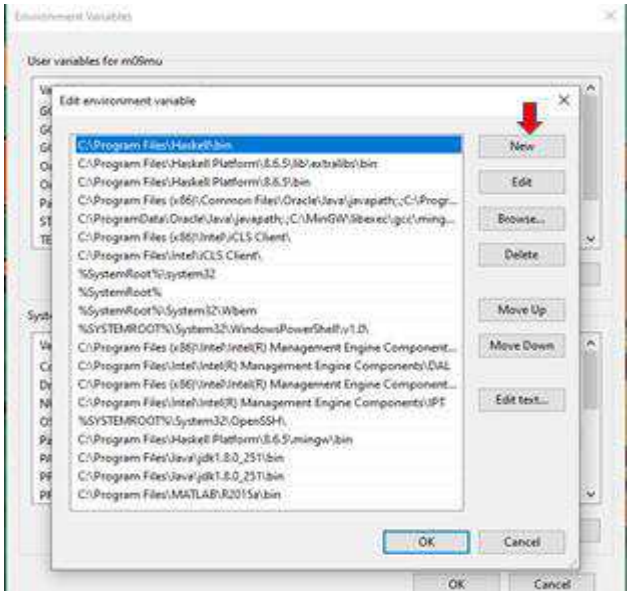
Step-4: Copy the bin path



Step-5: Set the system variables

Go to advanced system settings and open the system properties tab. Select the environment variables.

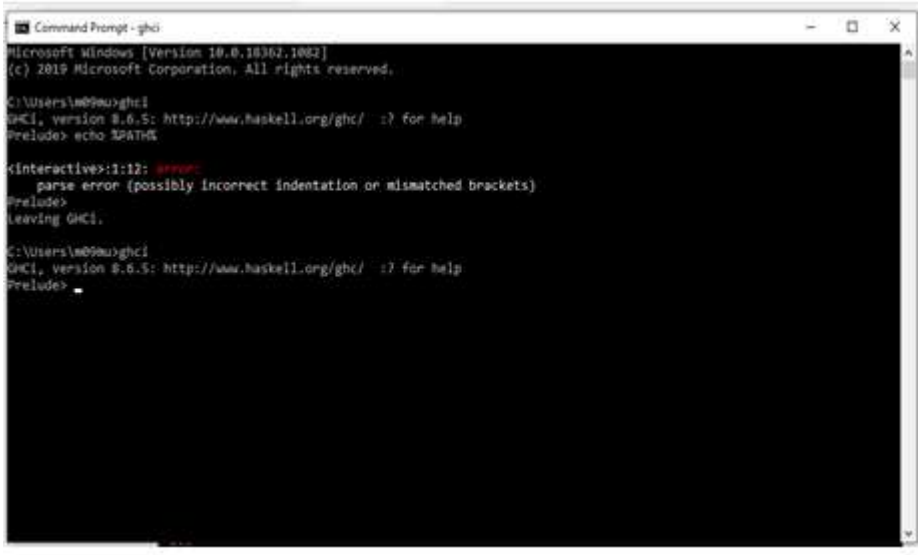




Set the system variables path as ‘C:\Program Files\Haskell Platform\bin’

Step-6: Check the compiler from command prompt

Open command prompt, at the prompt, type-‘ghci’. If the Haskell compiler is successfully installed, it will change the prompt to ‘Prelude’. You are now ready to work with Haskell



ghci is an interactive environment in which we can run haskell statsements as if they are commands.

Part 2: Perform basic arithmatic operations one by one in ghci and note the output:

- 2+3

- 2*3
- 2*(-3)
- 2-3
- 2/3
- 2 ^ 3
- it
- 50*100-4999
- 50*(100-4999)
- 40*100-3000+50/5
- (40*100-3000+50)/5

Part 3: Perform basic logical operations one by one in ghci and note the output:

- False
- True
- True && False
- True && True
- True || False
- not False
- not (True && False)

Part 4: Perform comparison operations one by one in ghci and note the output:

- 2 == 2
- 2 == 0
- 2 /= 0
- 2 /= 2
- 2 < 3
- 2 > 3
- “hi” == ”hi”
- “hi” == “Hi”

Post Experimental Exercise: (To be completed by students in writing as submission along with outputs of preceding section)

1. Explain Features of Haskell Programming Language

2. List basic data types in haskell

3. Differentiate between Imperative and functional programming language

Conclusion:

Thus we have learned how to install Haskell compiler and use GHCI for interactively exeuting haskell statements.

Reference:

1. Glasgow Haskell project home page, https://www.haskell.org/ accessed 30/11/2020.
2. Learn you Hakell for greate good. A biginner's guide, http://learnyouahaskell.com/
3. Michael L Scott, " Programming Language Pragmatics", Third edition, Elsevier publication

Experiment Number: 04

Aim: Tutorial on introduction to how to compile and execute Haskell code

Lab Objective: Design and implement declarative programs in functional and logic programming languages

Lab Outcome Mapped: Design and Develop solution based on declarative programming paradigm using functional and logic programming (LO2)

Requirements: Any text editor to be able to edit Haskell code and Glasgow Haskell Compiler 8.0+ version.

Theory:

Performance steps: