Name: Yash Mahajan Class:SE INFT B Roll No.-04 PID:191061

St. Francis Institute of Technology

Borivli (West), Mumbai-400103

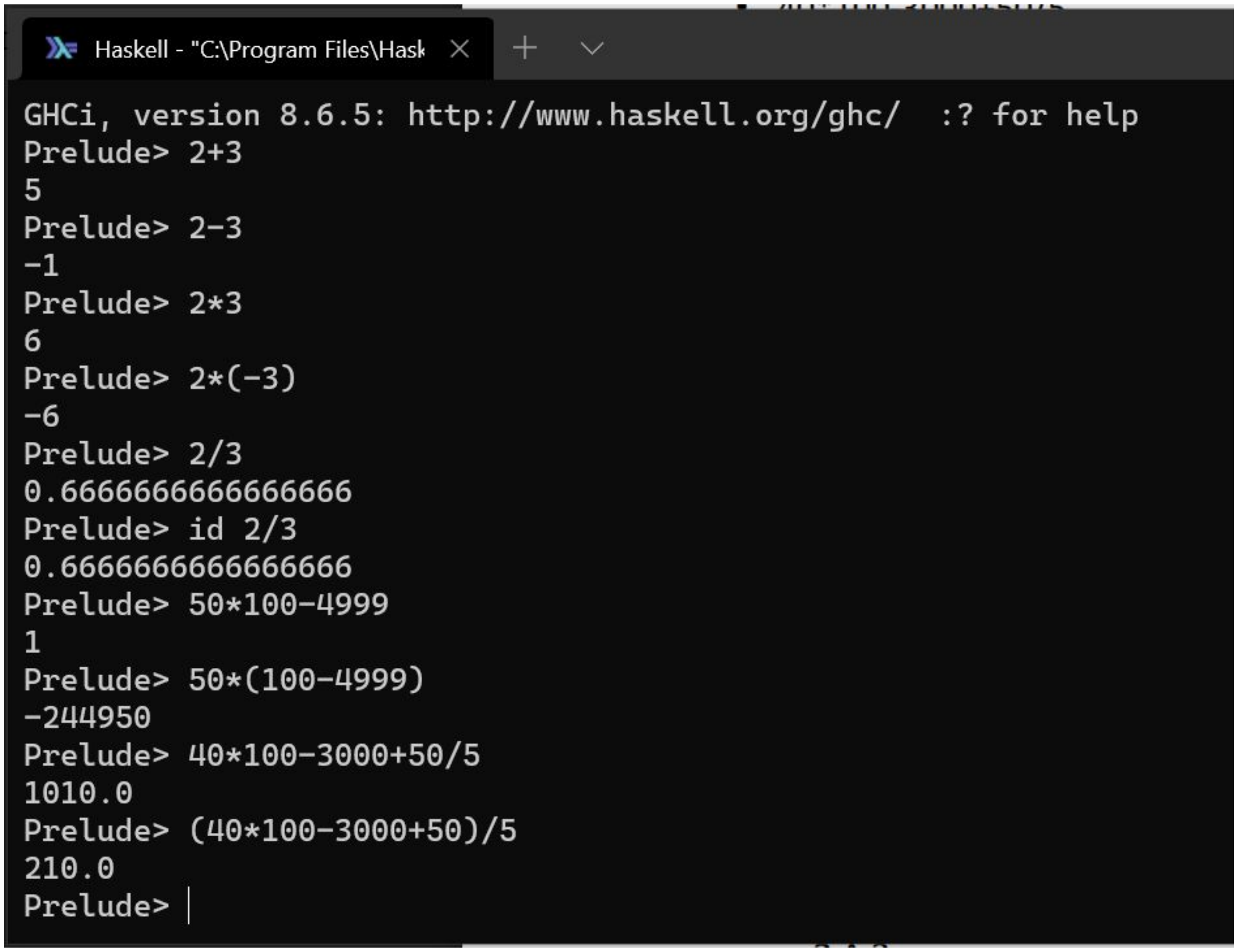
Department of Information Technology

Experiment – 4

1. Aim: Installation of Haskell Compiler and performing simple operations exercise

2. Program Code

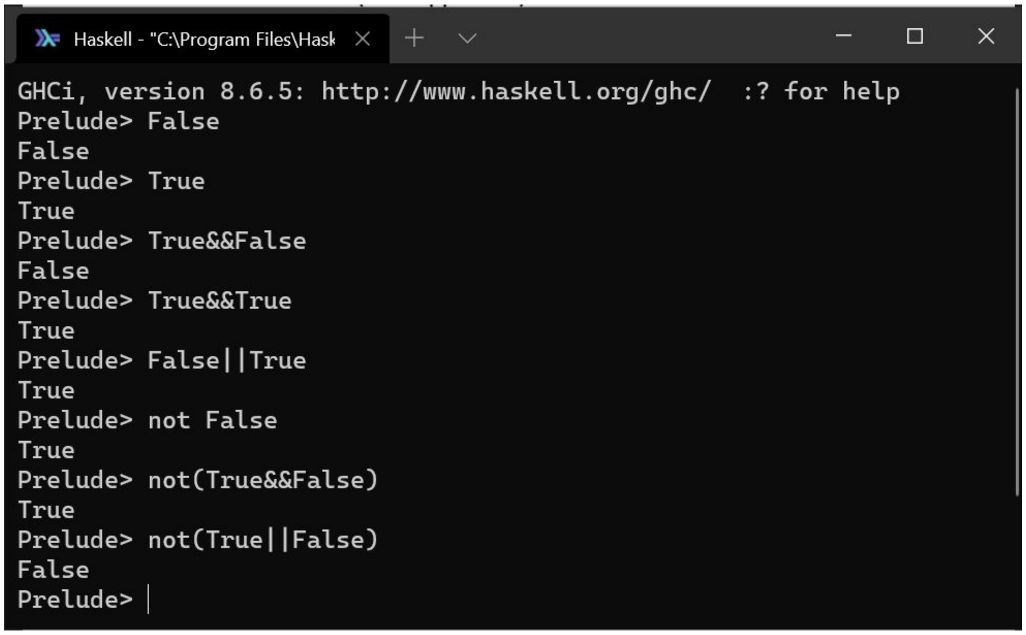
1. Arithmetic Operations



```
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
Prelude> 2+3
5
Prelude> 2-3
-1
Prelude> 2*3
6
Prelude> 2*(-3)
-6
Prelude> 2/3
0.6666666666666666
Prelude> id 2/3
0.6666666666666666
Prelude> 50*100-4999
1
Prelude> 50*(100-4999)
-244950
Prelude> 40*100-3000+50/5
1010.0
Prelude> (40*100-3000+50)/5
210.0
Prelude> 
```

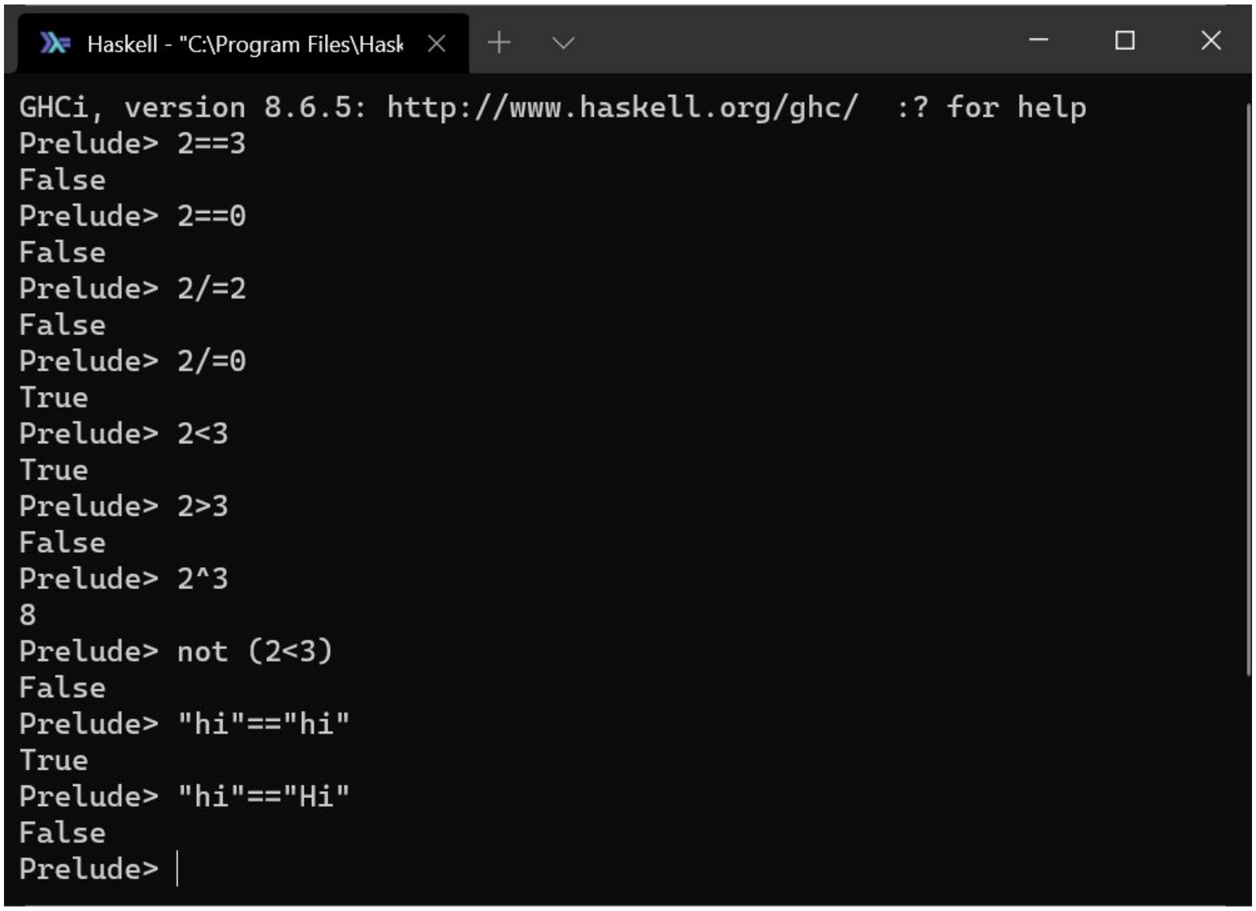
Name: Yash Mahajan Class:SE INFT B Roll No.-04 PID:191061

2. Logical Operations



```
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
Prelude> False
False
Prelude> True
True
Prelude> True&&False
False
Prelude> True&&True
True
Prelude> False||True
True
Prelude> not False
True
Prelude> not(True&&False)
True
Prelude> not(True||False)
False
Prelude>
```

3. Comparative Operations



```
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
Prelude> 2==3
False
Prelude> 2==0
False
Prelude> 2/=2
False
Prelude> 2/=0
True
Prelude> 2<3
True
Prelude> 2>3
False
Prelude> 2^3
8
Prelude> not (2<3)
False
Prelude> "hi"=="hi"
True
Prelude> "hi"=="Hi"
False
Prelude>
```

Name: Yash Mahajan Class:SE INFT B Roll No.-04 PID:191061

Post Experimental Exercise-

Questions

1. Differentiate between imperative and functional programming paradigm

Imperative Programming Paradigm	Functional Programming Paradigm
The paradigm consists of several statements, and after the execution of all of them, the result is stored.	Functional Programming is a programming paradigm that considers computation as the evaluation of mathematical functions and avoids changing state and mutable data.
The focus is on what needs to be done and how it should be done.	The focus is on what needs to be done rather than how it should be done, basically emphasize on what code code is actually doing.
Variables in imperative languages are inherently mutable in nature.	Variables in functional programming languages are immutable in nature.
Imperative languages primarily use loops, conditions and methods to move through the execution steps.	Functional programming languages rely mainly on function calls(including recursive calls) to achieve their desired goal.
Scala, Haskell and Lisp are functional programming languages.	C, C++, Java are imperative programming languages.

2. List basic data types in Haskell
The basic data types in Haskell are Bool, char, int, float, double, list and tuple.

3.Describe the features in Haskell
Haskell features lazy evaluation, lambda expressions, pattern matching, list comprehension, type classes and type polymorphism. It is a purely functional language, which means that functions generally have no side effects. A distinct construct exists to represent side effects, orthogonal to the type of functions. A pure function can return a side effect that is subsequently executed, modeling the impure functions of other languages. Haskell has a strong, static type system based on Hindley–Milner type inference. Its principal innovation in this area is type classes, originally conceived as a principled way to add overloading to the language,but since finding many more uses.

Conclusion:
Functional programming languages are specially designed to handle symbolic computation and list processing applications. In this experiment we used the prelude prompt to perform various operations. From this experiment we can infer that the Functional programming method focuses on results, not the process. Emphasis is on what is to be computed and variables are immutable.

D. References:

[1] https://www.haskell.org/
[2] http://learnyouahaskell.com/
[3] Michael L Scott, " Programming Language Pragmatics", Third edition, Elsevier publication