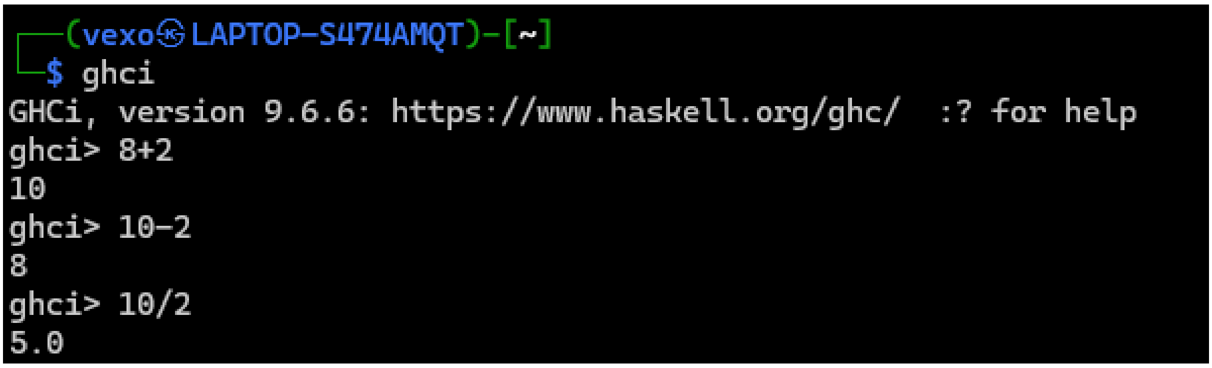
# 20CYS312 - Principles of Programing Languages - Lab Exercise 1

## 1. Basic Arithmetic:

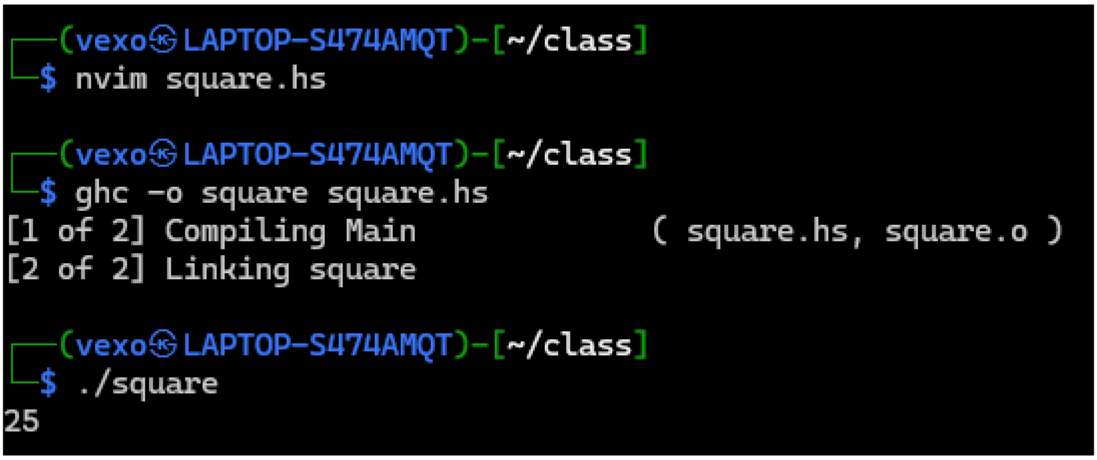
### Exercise 01 :Open GHCi and perform basic arithmetic operations:



```
┌──(vexo㉿LAPTOP-S474AMQT)-[~]
└─$ ghci
GHCi, version 9.6.6: https://www.haskell.org/ghc/  :? for help
ghci> 8+2
10
ghci> 10-2
8
ghci> 10/2
5.0
```

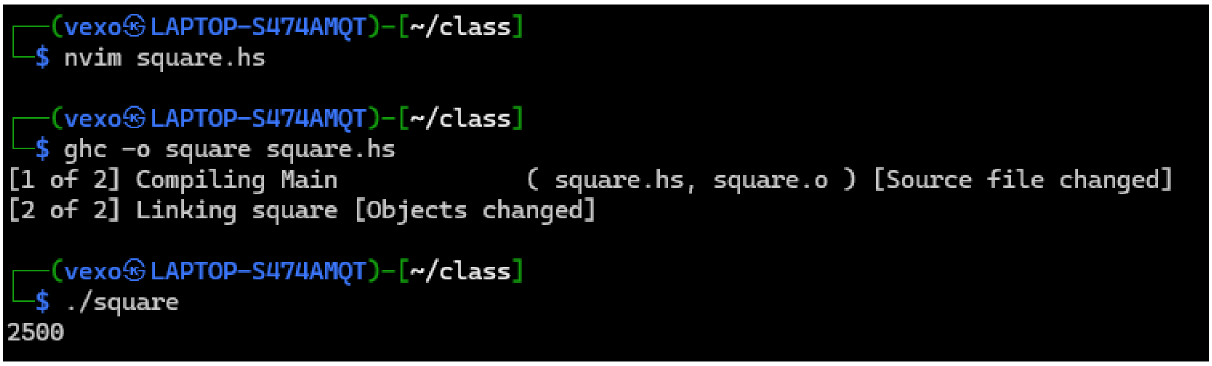
### Exercise 02 :Define a function to calculate the square of a number:

```haskell
square :: Int -> Int
square x = x * x

main :: IO ()
main = print (square 5)
```



```
┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ nvim square.hs

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ghc -o square square.hs
[1 of 2] Compiling Main             ( square.hs, square.o )
[2 of 2] Linking square

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ./square
25
```

```haskell
square :: Int → Int
square x = x * x

main :: IO ()
main = print (square 50)
```

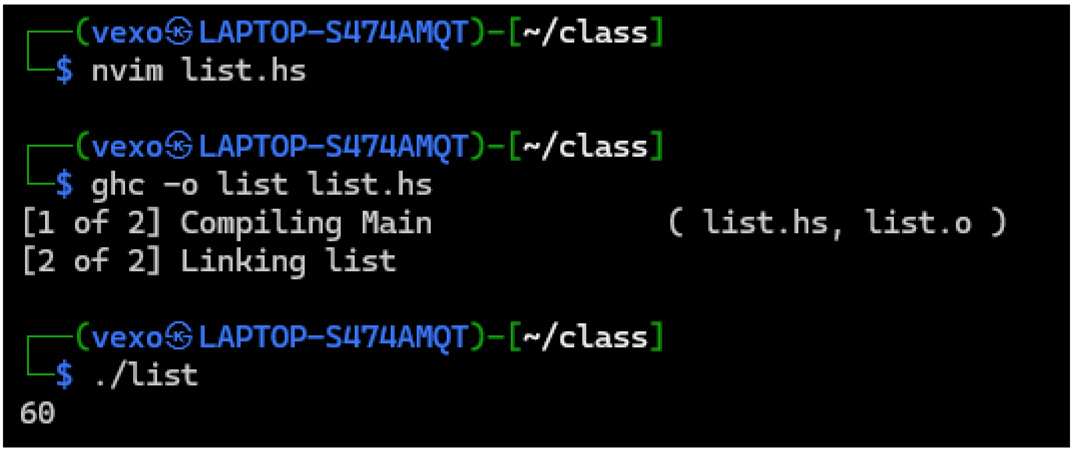


```
┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ nvim square.hs

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ghc -o square square.hs
[1 of 2] Compiling Main             ( square.hs, square.o ) [Source file changed]
[2 of 2] Linking square [Objects changed]

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ./square
2500
```

## 2. Defining and Using Lists:

### Exercise 3: Create a list of numbers and compute the sum of the list:

```haskell
sumList :: [Int] → Int
sumList [] = 0
sumList (x:xs) = x + sumList xs

main :: IO ()
main = print(sumList [10,20,30])
```

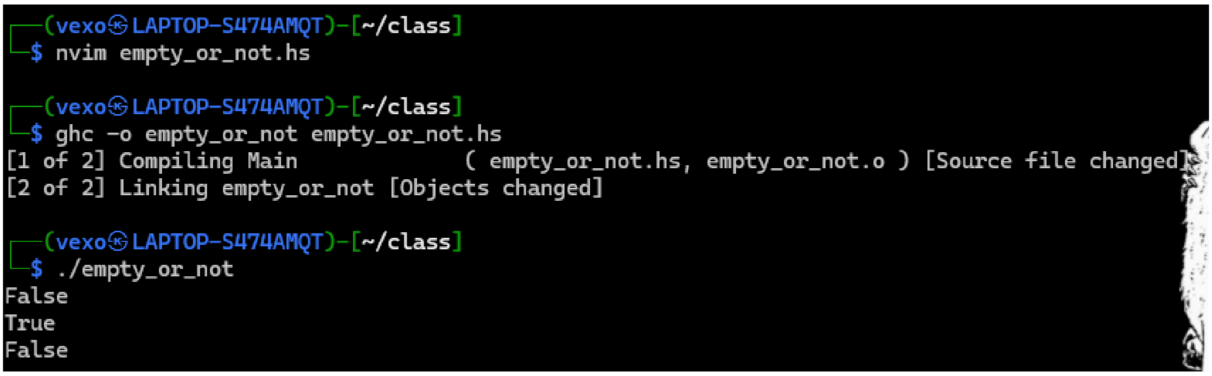


```
┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ nvim list.hs

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ghc -o list list.hs
[1 of 2] Compiling Main             ( list.hs, list.o )
[2 of 2] Linking list

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ./list
60
```

## 3. Pattern Matching with Lists:

### Exercise 4: Write a function to check if a list is empty:

```
isEmpty :: [a] → Bool
isEmpty [] = True
isEmpty _ = False

main :: IO ()
main = do
  print (isEmpty [10])
  print (isEmpty [])
  print (isEmpty [10,20,30])
```

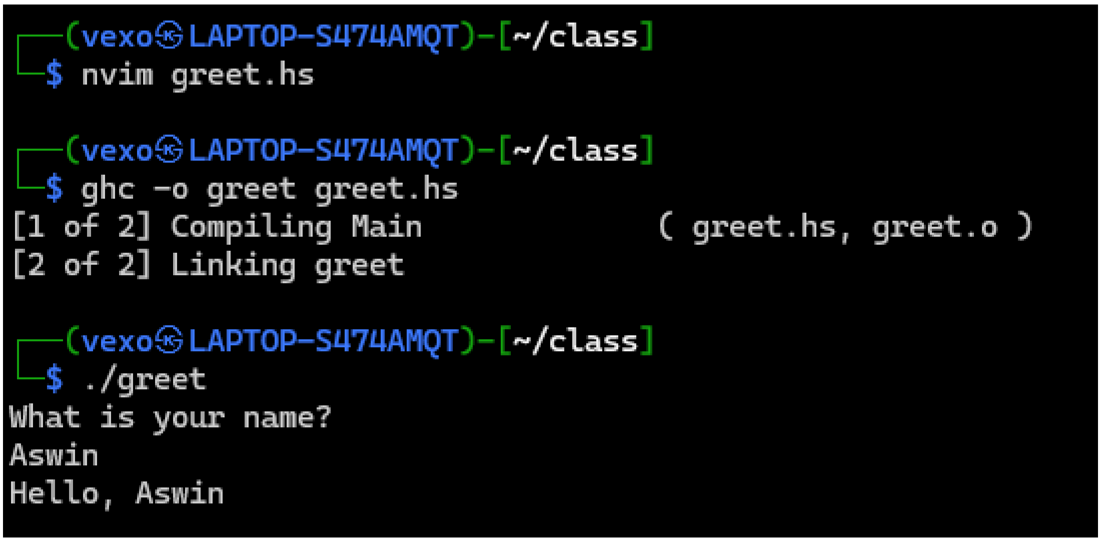


```
┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ nvim empty_or_not.hs

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ghc -o empty_or_not empty_or_not.hs
[1 of 2] Compiling Main             ( empty_or_not.hs, empty_or_not.o ) [Source file changed]
[2 of 2] Linking empty_or_not [Objects changed]

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ./empty_or_not
False
True
False
```

## 4. Simple IO Operations:

### Exercise 5: Write a program that asks the user for their name and prints a
greeting:

```
main :: IO ()
main = do
 putStrLn "What is your name?"
 name <- getLine
 putStrLn ("Hello, " ++ name)
```



```
┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ nvim greet.hs

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ghc -o greet greet.hs
[1 of 2] Compiling Main             ( greet.hs, greet.o )
[2 of 2] Linking greet

┌──(vexo㉿LAPTOP-S474AMQT)-[~/class]
└─$ ./greet
What is your name?
Aswin
Hello, Aswin
```