UNIT-III: Declarative Programming Paradigm: Functional Programming

Faculty In-charge

Mrinmoyee Mukherjee
Assistant Professor (IT Dept.)
email: mrinmoyeemukherjee@sfit.ac.in
Mob: 9324378409

Academic Year: 2023-24

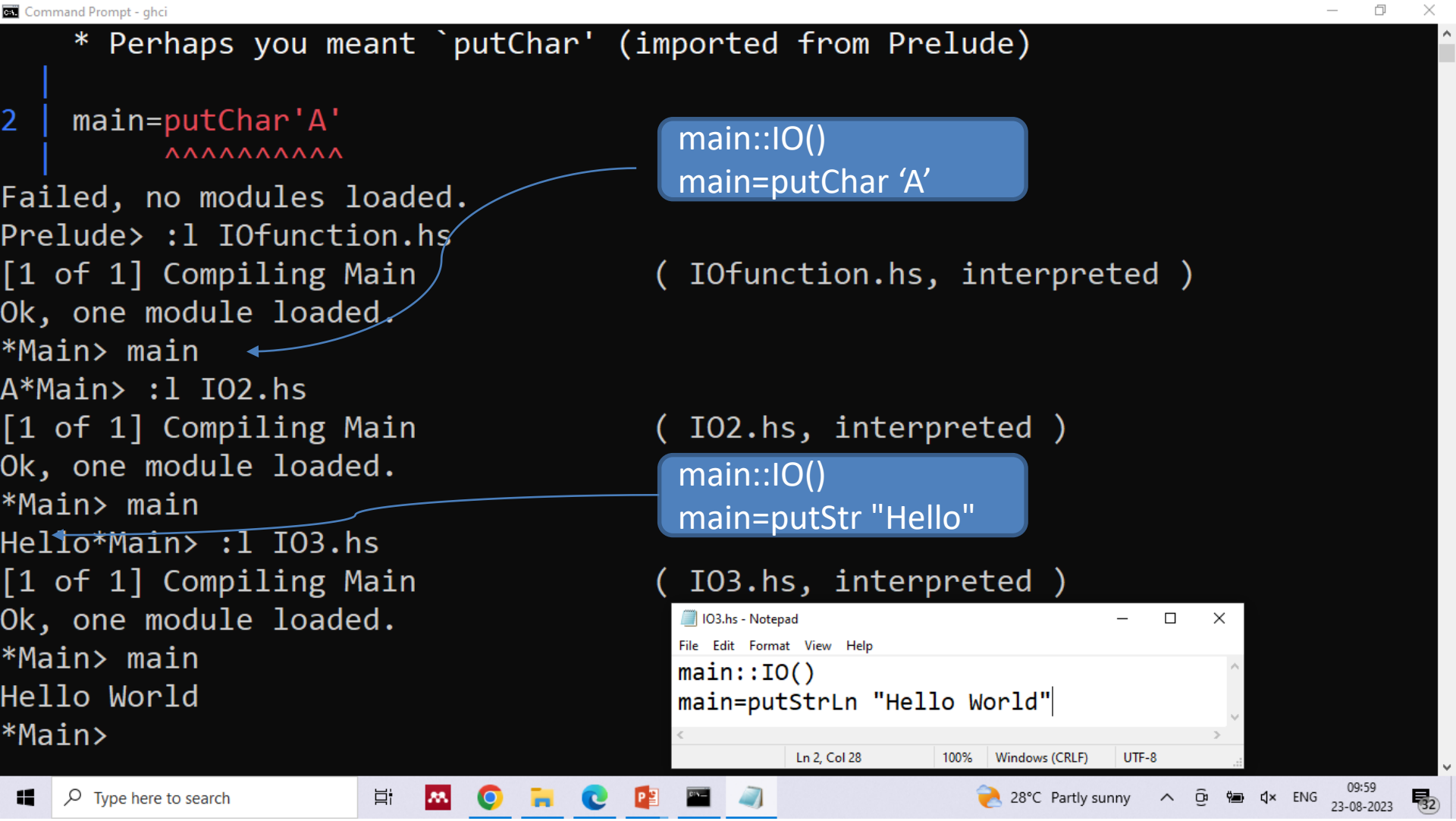
IO Actions

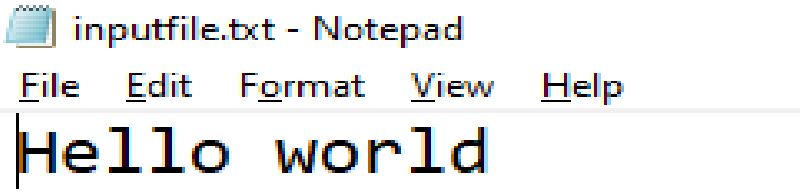
- **Input Actions:**
- These read from standard input (cin):
- **getChar** :: IO Char -- gets 1 char
- **getLine** :: IO String -- strips \n
- **getContents** :: IO String
- -- returns ALL the rest of cin!
- This reads the entire contents of a file!
- **readFile** :: FilePath -> IO String

```
main = do
  let file = "D:\abc.txt"
  a <- readFile file
  putStrLn a
```

- **Output Actions:**
- These write to standard output (cout)
- **putChar** :: Char -> IO ()
- **putStr** :: String -> IO ()
- **putStrLn** :: String -> IO () -- adds \n
- These write to (or create) output files:
- type FilePath = String
- **writeFile** :: FilePath -> String -> IO()
- **appendFile**:: FilePath -> String -> IO()

```
The first IO program is
main :: IO ()
main = putStrLn “Hello World”
```


Command Prompt - ghci
* Perhaps you meant `putChar' (imported from Prelude)
2 | main=putChar'A'
^^^^^^^^^^
Failed, no modules loaded.
Prelude> :l IOfunction.hs
[1 of 1] Compiling Main (IOfunction.hs, interpreted)
Ok, one module loaded.
*Main> main
A*Main> :l IO2.hs
[1 of 1] Compiling Main (IO2.hs, interpreted)
Ok, one module loaded.
*Main> main
Hello*Main> :l IO3.hs
[1 of 1] Compiling Main (IO3.hs, interpreted)
Ok, one module loaded.
*Main> main
Hello World
*Main>
main::IO()
main=putChar 'A'
main::IO()
main=putStr "Hello"
IO3.hs - Notepad
File Edit Format View Help
main::IO()
main=putStrLn "Hello World"
Ln 2, Col 28
100%
Windows (CRLF)
UTF-8
Type here to search
28°C Partly sunny
ENG
09:59
23-08-2023




```
Prelude> :l readfile.hs
[1 of 1] Compiling Main             ( readfile.hs, interpreted )
Ok, one module loaded.
*Main> main
Hello world
*Main> _
```

IO Program Ex-3

```
getInt p = do
{
print p;
line <- getLine;
return (read line :: Int)
}
```

```
getGreater p = do   {
 print p;
a <- getInt "First Num" ;
b <- getInt "Second Num" ;
print "greater num is ";
if a>=b then do
{   return (a)   }
else do
{    return (b)  }
}
main = do {
y <- getGreater "Program to find
greater num";
print y }
```