Lab Report

19CSE313 – Principles of Programming Languages

Tejas Gampawar   AM.EN.U4CSE22057

Criteria	Excellent	Good	Poor
Timely Submission			
Correctness of lab assignment			
Total Marks			
Signed By Lab Instructor			

Labsheet: 3
Date: 27/2/25
CO1: Demonstrate a comprehensive understanding of programming paradigms, with a focus on functional programming using Haskell and Scala

Question 1:Write a function even which takes an integer and returns True if even else False. Use Guarded equation.

<table>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">even_ :: Int -> Bool
even_ n | mod n 2 == 0 = True
      | otherwise = False</td><td>even_ 4</td><td>True</td></tr>
<tr><td>even_ 7</td><td>False</td></tr>
<tr><td>even_ 8</td><td>True</td></tr>
</table>

Question 2. Write a function abs to find the absolute of a number n.

<table>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">abs_ :: Int->Int
abs_ n | n < 0 = (-1 * n)
      | otherwise = n</td><td>abs_ 5</td><td>5</td></tr>
<tr><td>abs_ (-10)</td><td>10</td></tr>
<tr><td>abs_ 0</td><td>0</td></tr>
</table>

Question 3. Write a function max to find the largest among two numbers using guarded expressions.

Code	Input	Output
max_ :: Int-> Int -> Int max_ a b | (a>b) = a | otherwise = b	max_ 10 50	50
	max_ 5 (-10)	5
	max_ 5 5	5

Question 4. Write a function max3 to find the largest among three numbers using guarded expressions.

Code	Input	Output

max3_ :: Int->Int->Int->Int
max3_ a b c| (a >=b && a >=c) = a
         | (b >=c) = b
         | otherwise = c
	max3_ 10 20 30	30

	max3_ (-10) 20 0	20
	max3_ (-1) (-2) (-3)	-1

Question 5. Write a function to read a numeric value and return 1 if the number is positive else (-1).

Code	Input	Output
sign_ :: Int->Int sign_ n | n < 0 = (-1) | otherwise = 1	sign_ 10	1
	sign_ 0	1
	sign_ (-5)	-1

<table>
<tr><td colspan="3"><b>Question 6. Write a function changeCase, to convert a character from uppercase to lowercase character and vice versa.</b></td></tr>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">changeCase :: Char -> Char
changeCase x
 | (fromEnum(x) >= 65 && fromEnum(x) <= 91) = toEnum(fromEnum(x) + 32)::Char
 | (fromEnum(x) >= 97 && fromEnum(x) <= 123) = toEnum(fromEnum(x) - 32)::Char</td><td>changeCase 'A'</td><td>a</td></tr>
<tr><td>changeCase 'a'</td><td>A</td></tr>
<tr><td>changeCase 'Z'</td><td>z</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Question 7. Haskell function isVowel to check if a character is vowel or not.</b></td></tr>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">isVowel :: Char -> Bool
isVowel x
| (x == 'a') || (x == 'e') || (x == 'i') || (x =='o') || (x == 'u') = True
| (x == 'A') || (x == 'E') || (x == 'I') || (x =='O') || (x == 'U') = True
 | otherwise = False</td><td>isVowel 'A'</td><td>True</td></tr>
<tr><td>isVowel 'a'</td><td>True</td></tr>
<tr><td>isVowel 'B'</td><td>False</td></tr>
</table>

<table>
<tr><td colspan="3"><b>Question 8. Haskell function charType:: Char->String, to check if a character is a vowel or consonant.(use otherwise)</b></td></tr>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">charType :: Char -> String
charType a
| (a == 'a') || (a == 'i') || (a == 'e') || (a == 'o') || (a == 'u') = "Vowel"
| (x == 'A') || (x == 'E') || (x == 'I') || (x =='O') || (x == 'U') = "Vowel"
| otherwise = "Consonant"</td><td>charType 'A'</td><td>Vowel</td></tr>
<tr><td>charType 'B'</td><td>Consonant</td></tr>
<tr><td>charType 'i'</td><td>Vowel</td></tr>
</table>

<table>
<tr><th colspan="3">Question 9. Write Haskell Function to compare two values and any type and return the value of type Ordering. Use guarded conditions.</th></tr>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">compareAny :: Float -> Float -> Ordering
compareAny a b
  | (a > b) = GT
  | (a < b) = LT
  | otherwise = EQ</td><td>compareAny 10 20</td><td>LT</td></tr>
<tr><td>compareAny 10 0</td><td>GT</td></tr>
<tr><td>compareAny 10 10</td><td>EQ</td></tr>
</table>

<table>
<tr><th colspan="3">Question 10. Define bmicalculator, to find the bmi value using weight and height value and return a string based on the bmi value.use otherwise</th></tr>
<tr><th>Code</th><th>Input</th><th>Output</th></tr>
<tr><td rowspan="3">bmicalculator :: Int->Float->String
bmicalculator w h | ((w/ (h*h)) < 18.5) = "Underweight"
                  | ((w/ (h*h)) < 25) = "Normal"
                  | ((w/ (h*h)) < 30) = "Overweight"
                  | otherwise = "Underweight"</td><td>bmicalculator 50 1.65</td><td>Normal</td></tr>
<tr><td>bmicalculator 90 1.8</td><td>Overweight</td></tr>
<tr><td>bmicalculator 45 1.7</td><td>Underweight</td></tr>
</table>