

Roll No.: _____

Amrita Vishwa Vidyapeetham
Amrita School of Engineering, Coimbatore
B.Tech. Degree Examinations – May 2022

Sixth Semester

Computer Science and Engineering

19CSE313 Principles of Programming Languages

Duration: Three hours

Maximum: 100 Marks

CO	Course Outcomes
CO01	Understand and write pure functional programs (especially in Haskell and Scala).
CO02	Understand and write concurrent programs in Java.
CO03	Formulate abstractions with higher order procedures.
CO04	Formulate abstractions with data.

Answer all questions

- 1) Pangrams are strings that contain at least one occurrence of each letter of the English alphabet. They are not case-sensitive, so a letter may appear in either upper case, or lower case, or both. For example:
ghci> pangram "The quick brown fox jumps over a lazy dog."
True
Prelude> pangram "I wanna be a Pangram, can I? :)"
False
Write a Haskell function `pangram s`, which takes a string `s` as input, and returns `True` if `s` is a pangram, and `False` otherwise. [5] [CO01][BTL 3]
- 2) Write a Haskell expression in the form of a list comprehension to compute all perfect numbers. A number `x` with $x \geq 2$ is perfect if and only if the sum of its proper divisors is equal to itself. For example, 6 is perfect, since its proper divisors are 1, 2, and 3 and the sum of its proper divisors is 6. [5] [CO01][BTL 3]
- 3) Write a Haskell function ***applylistfun*** which takes a list of functions (*each of type $\ast \rightarrow \ast$*) and an item (*of type \ast*) and returns a list of the results of applying each function to the item. For example: `applylistfun [(+ 10), (* 3)] 2` will evaluate to `[12, 6]`. [5] [CO03][BTL 3]
- 4) Write a Haskell function ***lensort x***, which takes as input a *list x of sublists* and sorts each sublist in `x` based on the length. For example: `lensort [[1,2,3],[4,5],[6]]` must return `[[6],[4,5],[1,2,3]]` [5] [CO03][BTL 3]
- 5) There are 3-ways of classification of all natural numbers based on their aliquot sum. The aliquot sum of any integer $n > 0$ is defined as the sum of all factors of n , not including n itself. For example, the aliquot sum of 15 is $(1 + 3 + 5) = 9$, and the aliquot sum of 28 is $(1 + 2 + 4 + 7 + 14) = 28$. The number n is called deficient if its aliquot sum is smaller than n ; it is called perfect if its aliquot sum is n , and is called abundant if its aliquot sum is larger than n . Write a Haskell function ***classify n***, which takes an integer n as input, and returns its classification. The classification of any integer smaller than 1 can be stated as illegal. [5] [CO04][BTL 3]

- 6) Fill in the blanks in the following Haskell function **listaccess** that accepts a *list of values* and a *number* (list index) and returns the element in that index. Assume 0 indexing. If the index value exceeds the length of the list, the function should return an error message. For example, **listaccess [10, 11, 12, 13] 2** should return **12** as it is the 3rd element of the given list. [5] [CO04][BTL 2]

```
listaccess [] n = _____
listaccess (x:_) _ = _____
listaccess (_:_) _ = _____
```

- 7) Write a Scala function **find_max_product** to find maximum product of two integers in a given array of integers. For example if the input is nums = { 2, 3, 5, 7, -7, 5, 8, -5 }, the output is: Pair is (7, 8), Maximum Product: 56 [5] [CO01][BTL 3]
- 8) Write a Scala function to count the number of occurrences of each element in a given list. [5] [CO01][BTL 3]

- 9) Write a Scala function **compute**, to calculate the Hamming Distance between two DNA strands. Your body is made up of cells that contain DNA. Those cells regularly wear out and need replacing, which they achieve by dividing into daughter cells. In fact, the average human body experiences about 10 quadrillion cell divisions in a lifetime! When cells divide, their DNA replicates too. Sometimes during this process mistakes happen and single pieces of DNA get encoded with the incorrect information. If we compare two strands of DNA and count the differences between them we can see how many mistakes occurred. This is known as the "Hamming Distance". We read DNA using the letters C, A, G and T. Two strands might look like:

```
GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT
^ ^ ^ ^ ^ ^ ^^
```

They have 7 differences, and therefore the Hamming Distance is 7.

[5] [CO01][BTL 3]

- 10) Write a Scala function **dropTillTrue** that accepts a *list of numbers* as well as a *boolean function* and drops the elements in the list till they satisfy the given boolean function returning the remaining list of numbers. For example: dropTillTrue(List(1,2,3), ((x: Int) => x < 2)) must return List(2, 3) [5] [CO03][BTL 3]

- 11) Fill in the missing portions of the following Scala code, which implements higher-order function that returns a function using pattern matching to perform arithmetic operations. [5] [CO03][BTL 2]

```
object P18 {
  def mathOperation(____: ____): (____, ____)=> ____ = (____: ____, y: ____)=> {
    name match {
      case _____
      case _____
      case _____
      case _____
    }
  }
  def add: (____, ____)=> ____ = mathOperation("_____")
  def mul: (____, ____)=> ____ = mathOperation("_____")
  def div: (____, ____)=> ____ = mathOperation("_____")
  def sub: (____, ____)=> ____ = mathOperation("_____")
  def main(args: Array[String]) {
    println(add(10, 5));
    println(mul(10, 5));
    println(div(10, 5));
  }
}
```

```
println(sub(10, 5));
}}
```

12) Predict and explain the output of the following Scala code:

[5] [CO03][BTL 2]

```
object HelloWorld {
  def main(args: Array[String]) {
    println("Hello, world!")
    val m1 = List(3, 6, 2, 9, 21)
    val result = m1.forall(y => {y % 3 == 0})
    println(result)
  }
}
```

13) Write a Scala function '**compress**' to check if a list contains repeated elements and if so, they should be replaced with a single copy of the element. The order of the elements should not be changed.

[5] [CO04][BTL 3]

For Example:

```
scala> compress(List('a, 'a, 'a, 'a, 'b, 'c, 'c, 'a, 'a, 'd, 'e, 'e, 'e, 'e))
res0: List[Symbol] = List('a, 'b, 'c, 'a, 'd, 'e)
```

14) Define a Scala function that will identify whether a given numeric value is divisible by 2. Using this function, and starting from the left side to the right side of a list, remove all the elements which are factors of two. But, the function must stop removing the elements as soon as a number literal is not a factor of two.

[5] [CO04][BTL 3]

Example: For the Input: 2, 8, 19, 20, 25, 50, 100, 10 ; Output is: 19, 20, 25, 50, 100, 10.

15) Write a program in Scala to do the task of building a high-score component for the classic *Frogger* game. The scores obtained by a player are stored in a list. The task is to write functions – **latest**, **personalBest**, **personalTop** and **differenceOfBestReport** that return the **last added**, **highest (best)**, the **three highest scores** and the **difference between the best and the last** scores from the score list respectively. For example:

[5] [CO04][BTL 3]

For the input score list: List(2,1,3,5,4))

latest(List(2,1,3,5,4)) returns 4

personalBest(List(2,1,3,5,4)) returns 5

personalTop(List(2,1,3,5,4)) returns List(5, 4, 3)

differenceOfBestReport(List(2,1,3,5,4)) returns Your latest score was 4. That's 1 short of your personal best!

16) Write a Scala function **removeat** that accepts a list of numbers as well as the index (assuming list index starts from 1) and returns a list with the element at the given index removed. For example, removeat(List(1,2,3,4,5), 4) should return List(1,2,3,5).

[5] [CO04][BTL 3]

17)

a. Explain Fork/Join framework in Java [4 marks]

b. Write a Java program to calculate the factorial value of a number by creating a new thread. [6 marks]

[10] [CO02][BTL 2]

18)

a. Comment about the truth or falsity of the following statements.

[10] [CO02][BTL 2]

1. An application can be concurrent but not parallel [1 Mark]

2. An application can be parallel but not concurrent [1 Mark]

3. An application can be both parallel and concurrent [2 Marks]

4. An application can be neither parallel nor concurrent [1 Mark]

- b. How is concurrency achieved in Java/Scala/Haskell? (Choose any one language for your answer). [5 Marks]

Course Outcome /Bloom's Taxonomy Level (BTL) Mark Distribution Table

CO	Marks	BTL	Marks
CO01	25	BTL 1	0
CO02	20	BTL 2	35
CO03	25	BTL 3	65
CO04	30	BTL 4	0