

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING
Third Year, Second Semester Examination, 2018

Principles of Programming Languages

Time- Three Hours

Full Marks-100

Answer any five questions

1. (a) Discuss about different kinds of data abstraction techniques.
 (b) Describe Von Neumann bottleneck.
 (c) How can a programming language be defined? 10+2+8

2. (a) How would you define efficiency of a programming language?
 (b) Give an example of orthogonal language design.
 (c) Discuss about Turing tarpit.
 (d) State Flon's axiom. 10+4+3+3

3. (a) Write code snippets to compute factorial of a number following imperative, functional and logic programming paradigm.
 (b) Write referentially transparent code to compute gcd of a number. Justify your answer.
 (c) How can functions be treated as first class data values? Give an example. 10+6+4

4. (a) What is behavior parameterization? How is it implemented in Java?
 (b) Which of these lambda expressions are valid `Function<Long,Long>` implementations? Explain. (i) $x \rightarrow x + 1$; (ii) $(x) \rightarrow (y) \rightarrow (z) \rightarrow x + y + z + 1$;
 (c) Would the following Java code compile? Give reasons w.r.t functional interfaces.
 i. `Runnable helloWorld = () -> System.out.println("hello world");`
 ii. The lambda expression being used as an `ActionListener`:
`JButton button = new JButton();`
`String Name=getUserName();//Assume the method is defined`
`button.addActionListener(event->System.out.println("hello"+ name);`
 (d) Find the String with the largest number of lowercase letters from a `List<String>` using Lambda expressions in Java.
 (e) Given a text file, print the duplicate words using Lambda expressions in Java. 7+3+4+2+4

[Turn over

5. (a)

<pre>public class Trader private final String name; private final String city; public Trader(String n, String c); public String getName(); public String getCity();</pre>	<pre>public class Transaction private final Trader trader; private final int year; private final int value; public Trader getTrader(); public int getYear(); public int getValue();</pre>
--	--

Given the Trader and Transaction class (as shown), write code snippets using Java Streams API for the following queries:

- Find out a list of all the unique cities where the traders work.
 - Print the maximum and minimum of all transactions' values for the traders living in Cambridge.
 - Group transactions by cities and then further categorize by whether they're expensive or not (multilevel groupings should be used).
- (b) Write implementation of the functions max(), map(), filter() and count() using only reduce and Lambda expressions. You can return a List instead of a Stream if you want.

10+10

6. (a) Use normal order reduction and applicative order reduction to reduce the following Lambda expressions.

- $(\lambda x. \lambda z.z) ((\lambda y. yy) (\lambda u. uu))$
- $(\lambda x . x x x) (\lambda x . x x x)$
- $((((\lambda f.(\lambda g.(\lambda x.((fx)(g x)))))(\lambda m.(\lambda n.(n m)))))(\lambda n.z))p$

- How would you find predecessor and successor of 2 in Lambda calculus?
- How can the successor function be used to sum two natural numbers? Show the steps to find out the sum of 2 and 5. $(3+2+3)+8+4$

7. (a) Write Prolog clauses to express the relationships: grandparent, cousin, sibling and mother. Given Parent (X,Y) means X is a parent of Y.

- Write a program in Prolog to sort a list of numbers according to insertion sort algorithm.
- Given the following Prolog clauses:

```
ancestor(X, Y) :- parent(X, Z) , ancestor(Z, Y).
ancestor(X, X).
parent(amy, bob).
```

Place cut in such a way that (i) all solutions can be found; (ii) all solutions are pruned; (iii) one solution is found.

Show the search tree to be generated for the query ancestor(X, bob) for each of the cases. 6+5+9

8. (a) Describe multimethods with respect to object oriented programming.

- Compare between abstract methods and higher order functions.
- Describe currying in lambda calculus. How is it supported in Java through functional interfaces? Write relevant code snippets.
- Discuss width subtyping and depth subtyping and their relevance in object oriented programming. 4+4+7+5