

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING
Third Year, Second Semester Examination, 2017

Principles of Programming Languages

Time- Three Hours

Full Marks-100

Answer any five questions

1. (a) What are the different kinds of abstractions that can be applied on data?
 (b) What kind of abstractions is needed for parallel programming?
 (c) Discuss the properties of imperative programming paradigm. State its limitations.
 (d) Differentiate between procedures and functions. 7+3+6+4

2. (a) Describe the concept of regularity.
 (b) Give an example to show a feature that follows regularity.
 (c) Give another example that does not follow regularity. Justify.
 (d) State Flon's axiom.
 (e) Compare between static typing and dynamic typing. How are these features implemented in Java? 5+2+3+2+(5+3)

3. (a) Write code snippets to compute gcd following imperative, functional and logic programming paradigm.
 (b) What is referential transparency? Give an example of a function that cannot easily be made referentially transparent. Give reasons.
 (c) Define lambda expression. Is it type-safe? How is it different from *method* in Java? 9+4+7

4. (a) `public interface Function<T, R> {R apply(T t);}`
 What kind of lambda expressions would use this functional interface for writing a Calculator software?
 (b) Which of these lambda expressions are valid `Function<Long,Long>` implementations?
 (i) `x -> x + 1`; (ii) `(x, y) -> x + 1`; (iii) `x -> x == 1`;
 (c) Why are default methods necessary for functional interfaces in Java? Describe its resolution rules. Can you override `equals()` in a default method?
 (d) Convert the following code snippet to internal iteration:

```
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
int sum = 0;
for (int n : numbers) {
    if (n % 2 == 1) {
        int square = n * n;
        sum = sum + square;
    }
}
System.out.println(sum);
```

3+3+10+4

5. (a)

```

Dish
private final String name;
private final boolean vegetarian;
private final int calories;
private final Type type;

public Dish(String name, boolean vegetarian, int calories, Type type);
public String getName();
public boolean isVegetarian();
public int getCalories();
public Type getType();
public String toString();
public enum Type { MEAT, FISH, OTHER }

```

Given the Dish class (as shown), write code snippets for the following queries:

- (i) Find out a list of Meat, Fish and other dishes each grouped into low calorie and high calorie subgroups.
- (ii) What is the lowest calorie dish for each sub group (Meat, Fish and Other) in a menu (Menu is a class having Dish objects)?
- (iii) Count the total number of vegetarian dishes present in a menu.
- (b) From an array of words, (i) count and print the number of different words using Streams; (ii) print the sum of the length of all the words of length greater than 4; (iii) count and print the number of different letters.
- (c) Write a piece of code using streams that multiplies the numbers in a list together and adds the result with 25. Does it work sequentially? Can it execute with parallel streams? 7+9+4
6. (a) Use normal order reduction and applicative order reduction to reduce the following Lambda expressions.
- (i) $(\lambda x . (\lambda y z . z y) x) p (\lambda x . x)$
- (ii) $(\lambda x . x x x) (\lambda x . x x x)$
- (iii) $(\lambda c . c (\lambda a . \lambda b . b)) ((\lambda a . \lambda b . \lambda f . f a b) p q)$
- (b) Natural numbers can be computed in Lambda calculus with successor function $n =_{\text{def}} \lambda w y x . y(w y x)$.
- (i) Show the steps to find out successor of 4.
- (ii) How to compute sum of two natural numbers using the successor function? Show the steps for adding 3 with 4.
- (iii) Define predecessor function assuming the successor function is defined by S.
- (c) How is recursion supported in lambda calculus? Give an example. 6+(3+3+5)+3
7. (a) Write a program in Prolog to find the last element in a list. Justify your answer.
- (b) Write a program in Prolog to sort a list of numbers according to quick sort algorithm.
- (c) Given the following Prolog clauses:
- ```

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

```
- Show the search tree to be generated for the query ancestor(X, bob). Discuss the role of cut here.
- (d) Write Prolog clauses to express the relationship: cousin, sibling. Given parent (X, Y). 3+8+6+3

8. (a) Differentiate between object oriented programming and functional programming from the perspective of program design. Give an example application where each paradigm fits best.
- (b) Compare between abstract methods and higher order functions.
- (c) Name an object oriented language that does not need the concept of *interface*. Why?
- (d) (i) Write a program in Java that can perform any kind of value conversion. Show that it may act as a currency converter to convert from GBP to INR (say) and how can the same program convert from Fahrenheit to Celsius.
- (ii) Which programming paradigm is used here? Justify. 6+3+3+(5+3)