

# DSC512 Programming and Data Structures

Dr. Arun Cyril Jose

# Prerequisite

- Knowledge of any computer programming language

# Course Grading Policy

Evaluation type	Grade points
Mid Semester Exams (20 marks each)	40
Quiz/Assignment	20
End Semester Examination	40
Lab	30

# Objective

- Foundations of the practical implementation and usage of Algorithms and Data Structures.
- To ensure that the student evolves into a competent programmer capable of designing and analyzing implementations of algorithms and data structures for different kinds of problems.
- To expose the student to the algorithm analysis techniques, to the theory of reductions, and to the classification of problems into complexity classes like NP

# Outcomes

- Design and analyse programming problem statements
- Choose appropriate data structures and algorithms, understand the ADT/libraries, and use it to design algorithms for a specific problem.
- Understand the necessary mathematical abstraction to solve problems
- Comprehend and select algorithm design approaches in a problem specific manner.

# Syllabus

**Introduction:** Introduction to Data Structures and Algorithms, Review of Basic Concepts, Asymptotic Analysis of Recurrences. Randomized Algorithms. Randomized Quicksort, Analysis of Hashing algorithms.

**Algorithm Analysis Techniques** - Amortized Analysis. Application to Splay Trees. External Memory ADT - B-Trees. Priority Queues and Their Extensions: Binomial heaps, Fibonacci heaps, applications to Shortest Path Algorithms. Partition ADT: Weighted union, path compression, Applications to MST. Algorithm Analysis and Design Techniques.

**Dynamic Programming, Greedy Algorithms**-Bellman-Ford. Network Flows-Max flow, min-cut theorem, Ford-Fulkerson, Edmonds-Karp algorithm, Bipartite Matching.

**Intractable Problems:** Polynomial Time, class P, Polynomial Time Verifiable Algorithms, class NP, NP completeness and reducibility, NP Hard Problems, NP completeness proofs, Approximation Algorithms.

# Learning Resources

1. Introduction to Algorithms, by T. H. Cormen, C. E. Lieserson, R. L. Rivest, and C. Stein, Third Edition, MIT Press.
2. Fundamentals of Data Structures in C by Horowitz, Sahni, and Anderson-Freed, Universities Press
3. Algorithms, by S. Dasgupta, C. Papadimitrou, U Vazirani, Mc Graw Hill.
4. Algorithm Design, by J. Klienbergl and E. Tardos, Pearson Education Limited.

# Questions



# Question 1

- When was the last time you have done a complete program i.e. did everything from scratch??
  - A. Always try to do it from scratch.
  - B. Why should I, when there are readymade solutions out there.
  - C. Long time ago.
  - D. It is not my strong suit.

## Question 2

- How would you rate your computer programming capabilities??
  - A. Excellent.
  - B. Good.
  - C. I can handle it.
  - D. Not my strong suit.

# Question 3

- Are you familiar with any of these terms: methods, functions, subroutines??
  - A. I know what all of them are.
  - B. I know what functions are.
  - C. I know what functions and subroutines are.
  - D. I cannot remember what any of them are.

# Question 4

- When was the last time you did programming in C language??
  - A. I do it as part of my job all the time.
  - B. Only did it during my Undergraduate days.
  - C. Had professional C programming experience.
  - D. No C programming language experience what so ever.

# Question 5

- On a scale from 1 to 10, how do you estimate your programming experience?? (*1: being very inexperienced to 10: very experienced*)
  - A. Between 3 and 1.
  - B. Between 4 and 6.
  - C. Over 7.
  - D. I do not want to answer this question.

# Question 6

- How many years have you been programming?
- A. No experience
- B. Around 1-3 Years.
- C. Around 5 Years.
- D. More than 7 Years.

# Question 7

• What will the following program do??

- A. Sort and print the array in ascending order.
- B. Sort and print the array in descending order.
- C. Print all the elements of the array.
- D. Does nothing.

```
1 public class Class1 {  
2     public static void main(String[] args) {  
3         int array[] = {14,5,7};  
4         for (int counter1 = 0; counter1 < array.length;  
5             counter1++) {  
6             for (int counter2 = counter1; counter2 > 0;  
7                 counter2--) {  
8                 if (array[counter2 - 1] > array[counter2]) {  
9                     int variable1 = array[counter2];  
10                    array[counter2] = array[counter2 - 1];  
11                    array[counter2 - 1] = variable1;  
12                }  
13            }  
14            for (int counter3 = 0; counter3 < array.length;  
15                counter3++)  
16                System.out.println(array[counter3]);  
17        }  
18    }
```

# Question 8

- How extensively have you learned Computer Science during your undergraduate days?
  
- A. All throughout my course.
- B. May be one or two Semesters.
- C. Not even a single Semester.
- D. I cannot remember.