

## COURSE DESCRIPTION

Dept., Number	CS331	Course Title	Data Structures and Algorithms
Semester hours	3	Course Coordinator	Dr. Gruia Calinescu, Associate Professor

### Current Catalog Description

Implementation and application of the essential data structures used in computer science. Analysis of basic sorting and searching algorithms and their relationship to these data structures. Particular emphasis is given to the use of object-oriented design and data abstraction in the creation and application of data structures. Prerequisite: CS 116 or CS 201. (2-2-3)

### Textbook

Teacher Supplied Material - <http://dijkstra.cs.iit.edu/cs331-sp08/schedule/>

### References

<http://dijkstra.cs.iit.edu/cs331-sp08/resources/>

### Course Outcomes

Students should be able to:

- Explain, implement, and apply the following data-structures:
  - lists (unordered and ordered), stacks, queues, expression trees, binary search trees, heaps, and hash tables.
- Analyze the time and space complexity of algorithms using asymptotic upper bounds (big-O notation).
- Explain and use references and linked structures.
- Outline basic object-oriented design concepts: composition, inheritance, polymorphism.
- Write and test recursive procedures, and explain the run-time stack concept.
- Analyze searching and sorting algorithms, and explain their relationship to data-structures.
- Choose and implement appropriate data-structures to solve an application problem.
- Explain how to use unit tests and version control in your software development.

### Relationship between Course Outcomes and Program Outcomes

The following Program Outcomes are supported by the above Course Outcomes:

- a. An ability to apply knowledge of computing and mathematics appropriate to the discipline

- b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- c. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
- d. An ability to function effectively on teams to accomplish a common goal
- i. An ability to use current techniques, skills, and tools necessary for computing practices.
- j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
- k. An ability to apply design and development principles in the construction of software systems of varying complexity

#### Prerequisites by Topic

CS 116 or CS 201 - Experience in object-oriented programming

#### Major Topics Covered in the Course

1. Abstraction/Variables	3 hours
2. Linux/Subversion	3 hours
3. Lists (Array and Linked List)	7.5 hours
4. Stacks and Queues	6 hours
5. Ordered Lists, Sorting	7.5 hours
6. Doubly-Linked Lists	4.5 hours
7. Binary Search Trees	6 hours
8. Expression Trees	3 hours
9. Heaps	4.5 hours
10. Hash Tables	6 hours
11. Project(s) discussion, Midterm(s) and discussion, Project(s) evaluation	9 hours
Final Exam	-
	60 hours

#### Assessment Plan for the Course

End of every semester Course Objective Assessments by CS department. End of semester Course Evaluations by IIT. Reviewed every Spring semester by CS Undergraduate Studies Committee for possible updates in the following Fall. Once every 4-5 years a detailed review of all materials for the course is made by the CS Undergraduate Studies Committee.

How Data in the Course is Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

See the assessment discussion under Criterion 4
---

*For a computer science program*

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	.33		Software design	.33	
Data structures	1.5	.5	Concepts of programming languages	.33	