Welcome!

CS1000
Explorations in Computing
Department of Computer Science
Michigan Technological University

Dr. Nilufer Onder
Fall 2015
Fisher 139

Outline

- Information about me
- Tips to connect with faculty
- Course information
- Computer Science curricula

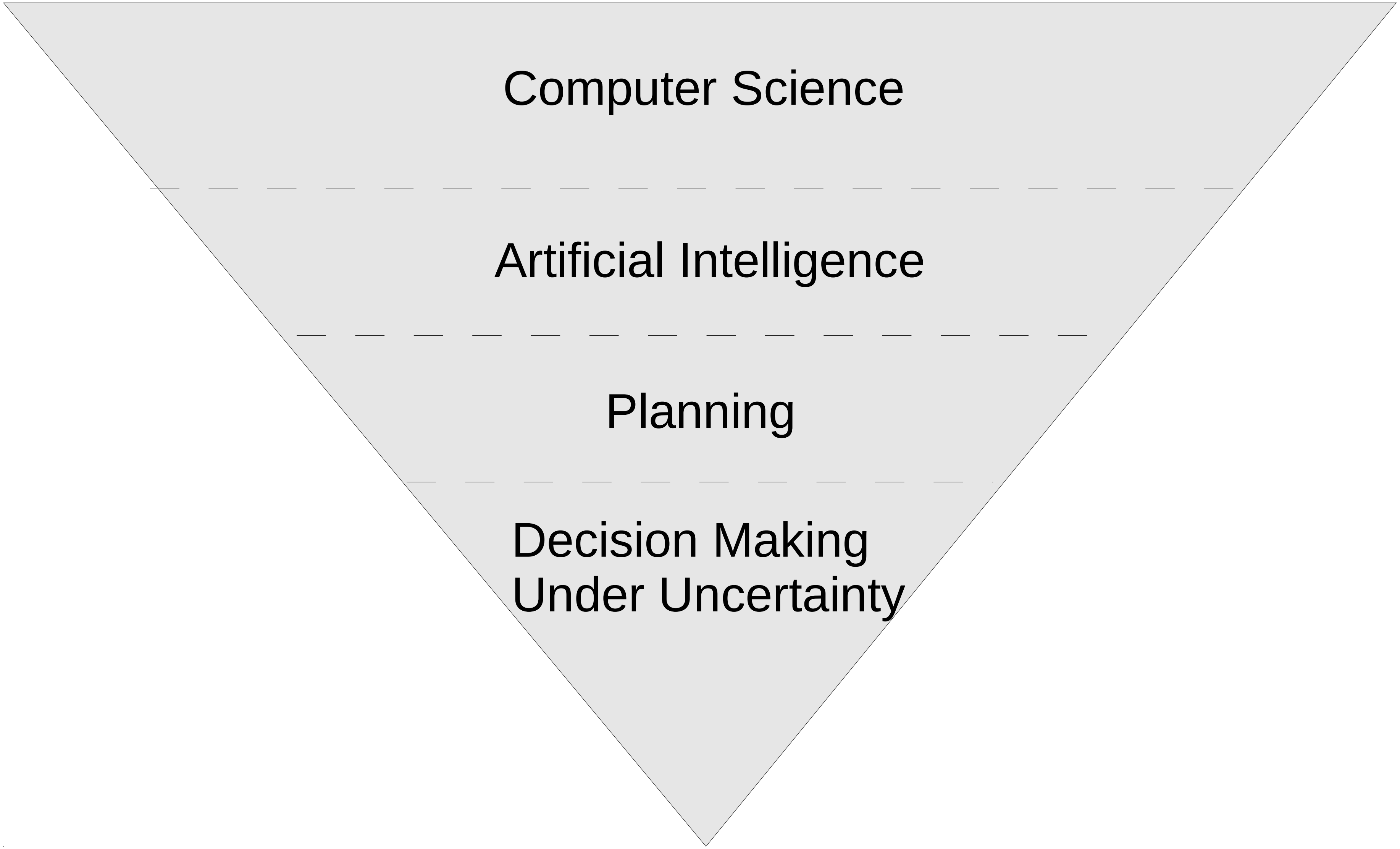
Short biography

- BSc in Computer Engineering
  Orta Dogu Teknik Universitesi
- MSc in Computer Engineering
  Orta Dogu Teknik Universitesi
- Worked as a systems analyst
- PhD in Computer Science
  University of Pittsburgh
- Came to Michigan Tech in 1999

Who motivated me

- My parents and family
- Faculty, advisors, bosses

Research Area



Computer Science

Artificial Intelligence

Planning

Decision Making
Under Uncertainty

Courses I teach

- CS1000 – Explorations in Computing
  undergrad, required
- CS 3311 - Formal Models of Computation
  undergrad, required
- CS 4811 – Artificial Intelligence
  undergrad, elective
- CS 5811 – Advanced Artificial Intelligence
  grad
- SSE 3200, CS 3090 – Web Based Services
  undergrad

Outline

- Information about me (done)
- Tips to connect with faculty
- Course information
- Computer Science curricula

Tips to connect with faculty

- Don't hesitate to initiate conversations with your professors
- Lots of professional advantages to getting to know each other
- Logistics, scheduling

Outline

- Information about me (done)
- Tips to connect with faculty (done)
- Course information
- Computer Science curricula

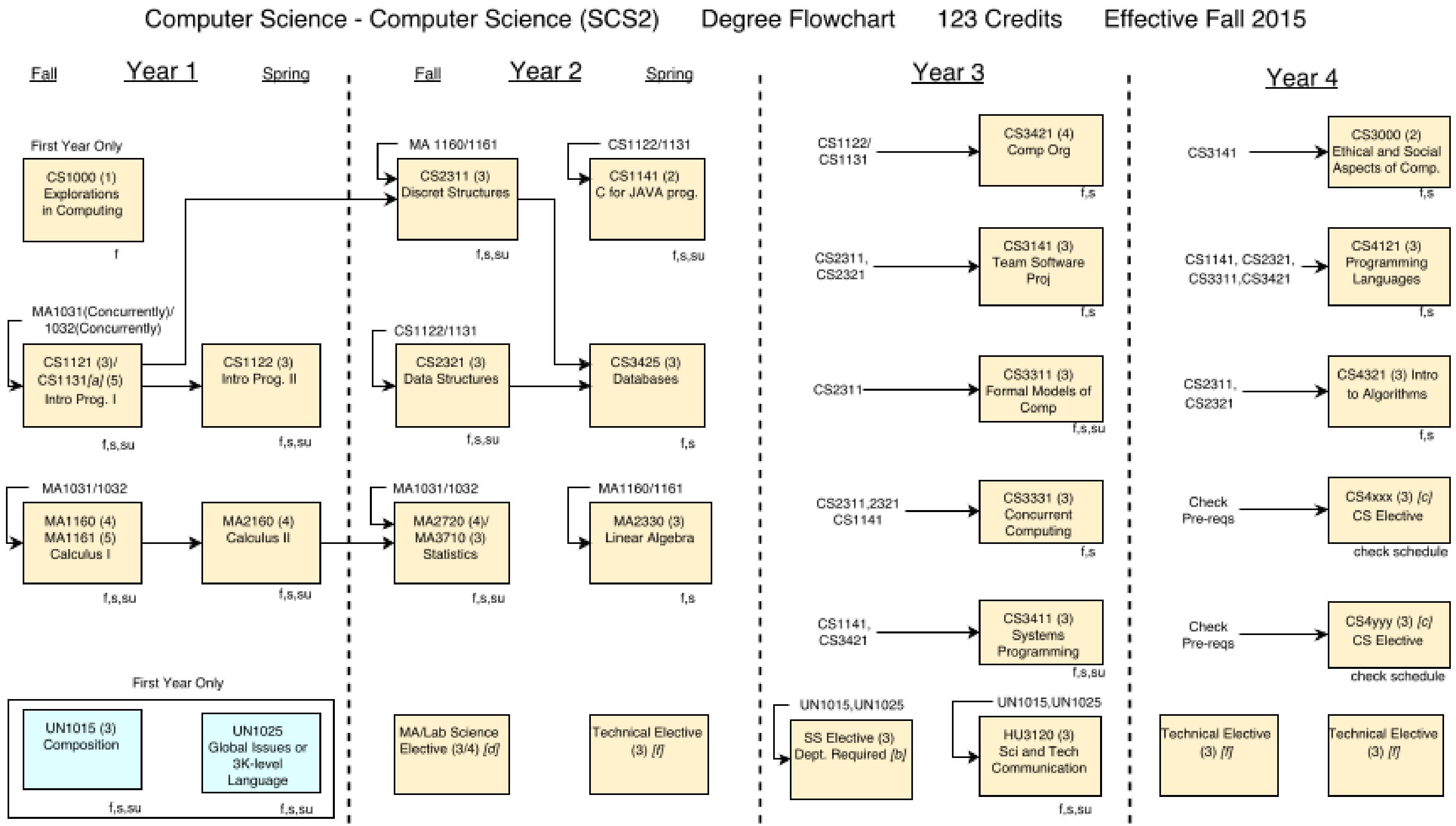
CS1000

- “Explorations” in Computing
- Explorations that lead to success in
  - Academic life
  - Career planning
- Forward looking course
- Check out the course syllabus

Outline

- Information about me (done)
- Tips to connect with faculty (done)
- Course information (done)
- Computer Science curricula

Computer science degree flowchart



Computer Science - Computer Science (SCS2) Degree Flowchart 123 Credits Effective Fall 2015

Year 1: Fall / Spring

First Year Only
CS1000 (1) Explorations in Computing — f

MA1031(Concurrently)/ 1032(Concurrently)
CS1121 (3)/ CS1131[a] (5) Intro Prog. I — f,s,su
CS1122 (3) Intro Prog. II — f,s,su

MA1031/1032
MA1160 (4) MA1161 (5) Calculus I — f,s,su
MA2160 (4) Calculus II — f,s,su

First Year Only
UN1015 (3) Composition — f,s,su
UN1025 Global Issues or 3K-level Language — f,s,su

Year 2: Fall / Spring

MA 1160/1161
CS2311 (3) Discret Structures — f,s,su

CS1122/1131
CS1141 (2) C for JAVA prog. — f,s,su

CS1122/1131
CS2321 (3) Data Structures — f,s,su

CS3425 (3) Databases — f,s

MA1031/1032
MA2720 (4)/ MA3710 (3) Statistics — f,s,su

MA1160/1161
MA2330 (3) Linear Algebra — f,s

MA/Lab Science Elective (3/4) [d]

Technical Elective (3) [f]

Year 3

CS1122/ CS1131 → CS3421 (4) Comp Org — f,s

CS2311, CS2321 → CS3141 (3) Team Software Proj — f,s

CS2311 → CS3311 (3) Formal Models of Comp — f,s,su

CS2311,2321 CS1141 → CS3331 (3) Concurrent Computing — f,s

CS1141, CS3421 → CS3411 (3) Systems Programming — f,s,su

UN1015,UN1025
SS Elective (3) Dept. Required [b]

UN1015,UN1025
HU3120 (3) Sci and Tech Communication — f,s,su

Year 4

CS3141 → CS3000 (2) Ethical and Social Aspects of Comp. — f,s

CS1141, CS2321, CS3311,CS3421 → CS4121 (3) Programming Languages — f,s

CS2311, CS2321 → CS4321 (3) Intro to Algorithms — f,s

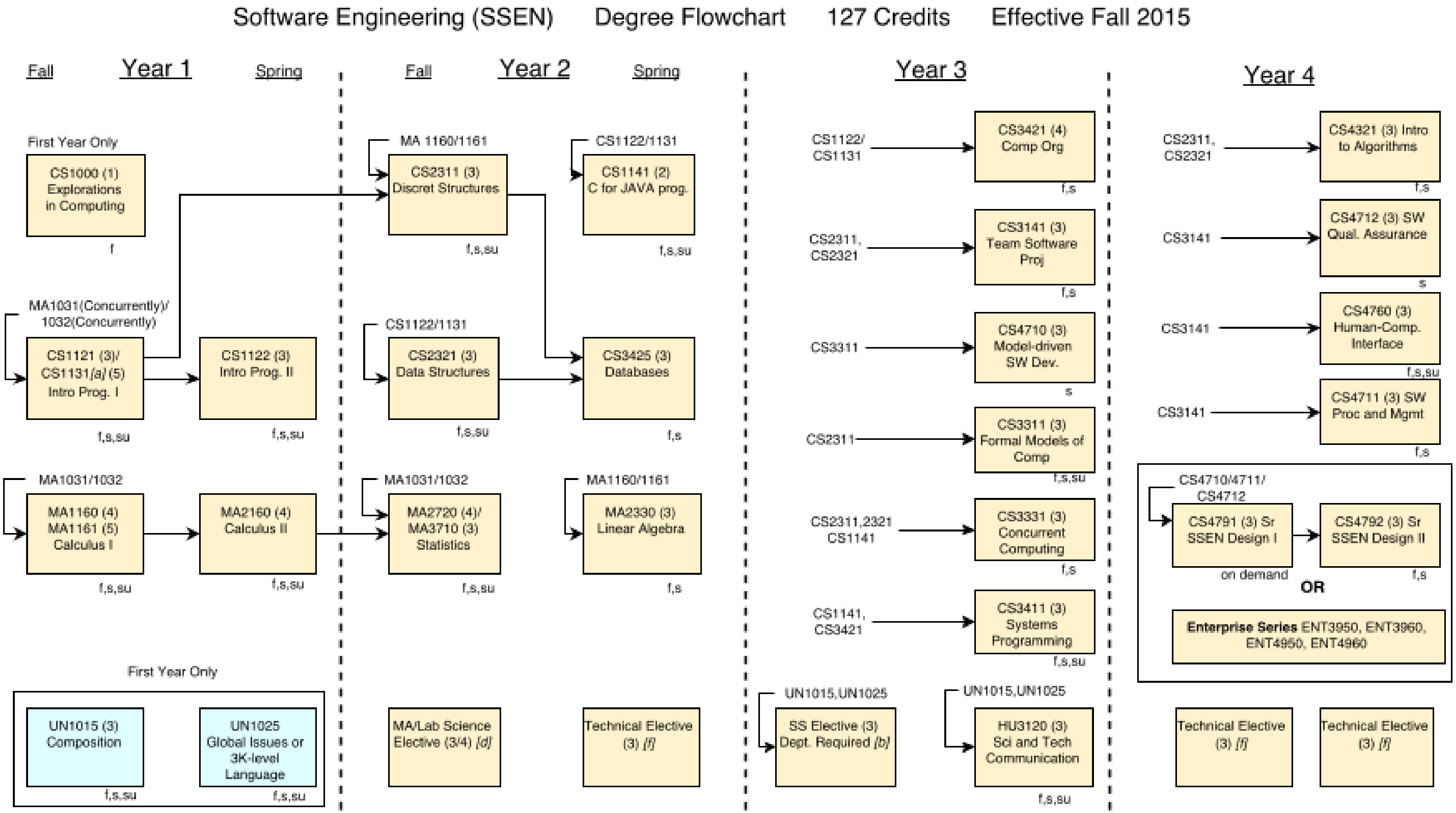
Check Pre-reqs → CS4xxx (3) [c] CS Elective — check schedule

Check Pre-reqs → CS4yyy (3) [c] CS Elective — check schedule

Technical Elective (3) [f]

Technical Elective (3) [f]

Software engineering degree flowchart



Software Engineering (SSEN) Degree Flowchart 127 Credits Effective Fall 2015

Year 1

Fall

First Year Only

CS1000 (1) Explorations in Computing
f

MA1031(Concurrently)/ 1032(Concurrently)

CS1121 (3)/ CS1131*[a]* (5) Intro Prog. I
f,s,su

MA1031/1032

MA1160 (4) MA1161 (5) Calculus I
f,s,su

Spring

CS1122 (3) Intro Prog. II
f,s,su

MA2160 (4) Calculus II
f,s,su

First Year Only

UN1015 (3) Composition
f,s,su

UN1025 Global Issues or 3K-level Language
f,s,su

Year 2

Fall

MA 1160/1161

CS2311 (3) Discret Structures
f,s,su

CS1122/1131

CS2321 (3) Data Structures
f,s,su

MA1031/1032

MA2720 (4)/ MA3710 (3) Statistics
f,s,su

MA/Lab Science Elective (3/4) *[d]*

Spring

CS1122/1131

CS1141 (2) C for JAVA prog.
f,s,su

CS3425 (3) Databases
f,s

MA1160/1161

MA2330 (3) Linear Algebra
f,s

Technical Elective (3) *[f]*

Year 3

CS1122/ CS1131 → CS3421 (4) Comp Org
f,s

CS2311, CS2321 → CS3141 (3) Team Software Proj
f,s

CS3311 → CS4710 (3) Model-driven SW Dev.
s

CS2311 → CS3311 (3) Formal Models of Comp
f,s,su

CS2311,2321 CS1141 → CS3331 (3) Concurrent Computing
f,s

CS1141, CS3421 → CS3411 (3) Systems Programming
f,s,su

UN1015,UN1025

SS Elective (3) Dept. Required *[b]*

UN1015,UN1025

HU3120 (3) Sci and Tech Communication
f,s,su

Year 4

CS2311, CS2321 → CS4321 (3) Intro to Algorithms
f,s

CS3141 → CS4712 (3) SW Qual. Assurance
s

CS3141 → CS4760 (3) Human-Comp. Interface
f,s,su

CS3141 → CS4711 (3) SW Proc and Mgmt
f,s

CS4710/4711/ CS4712

CS4791 (3) Sr SSEN Design I
on demand

CS4792 (3) Sr SSEN Design II
f,s

OR

Enterprise Series ENT3950, ENT3960, ENT4950, ENT4960

Technical Elective (3) *[f]*

Technical Elective (3) *[f]*

Question

- Where do curricula come from?

Computer Science Curricula 2013

Curriculum Guidelines for Undergraduate Degree Programs in Computer Science

December 20, 2013

The Joint Task Force on Computing Curricula
Association for Computing Machinery (ACM)
IEEE Computer Society

A Cooperative Project of

acm Association for Computing Machinery
Advancing Computing as a Science & Profession

IEEE

IEEE computer society

What to consider when designing a curriculum?

- Reflects the state of the art body of disciplinary knowledge (reasonable size)
- Is rigorous
- Is flexible to meet needs of individual departments and students
- Is pedagogically sound and complete
- Has good breadth and depth coverage
- Considers input and feedback from a broad community
- Revised continually

Characteristics of CS graduates

- Technical understanding of computer science
- Familiarity with common themes and principles
- Appreciation of the interplay between theory and practice
- System-level perspective
- Awareness of the broad applicability of computing

Characteristics of CS graduates (cont'd)

- Problem solving skills
- Project experience
- Commitment to life-long learning
- Commitment to professional responsibility
- Communication and organizational skills
- Appreciation of specific knowledge in other domains (cross disciplinary)

Knowledge areas and core hours

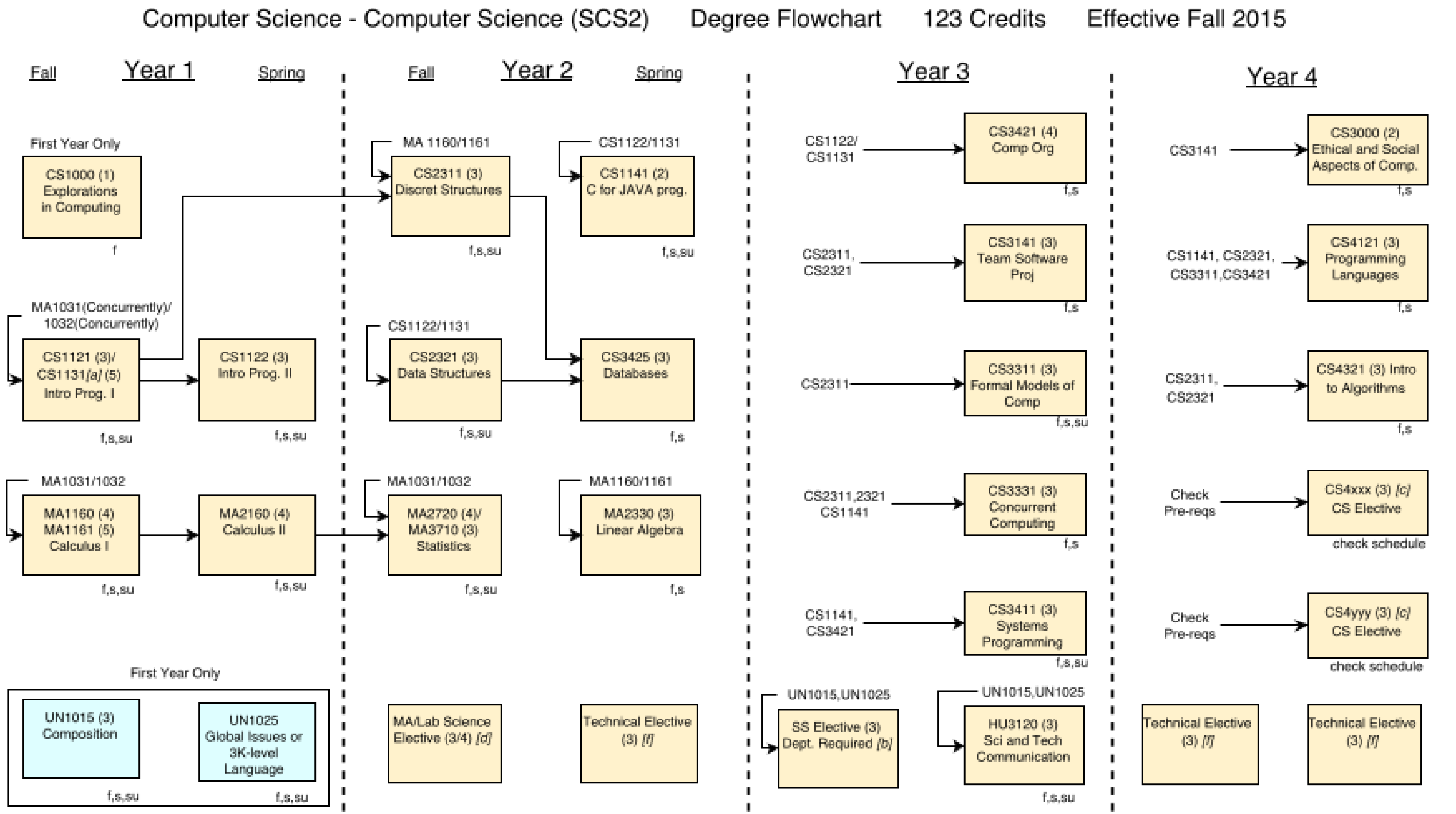
- Technologies change rapidly over time
- Essential concepts, perspectives, and methodologies that are constant define computer science
- The body of knowledge is organized into 18 knowledge areas. For each
  - Tier 1: essential for all CS programs
  - Tier 2: individual programs choose their coverage
- Knowledge areas are not intended to describe specific courses

Knowledge Area	**Tier 1**	**Tier 2**	**Total**
Algorithms and complexity	19	9	28
Architecture and Organization	0	16	16
Computational Science	1	0	1
Discrete Structures	37	4	41
Graphics and Visualization	2	1	3
Human-computer interaction	4	4	8
Information assurance and security	3	6	9
Information management	1	9	10
Intelligent systems	0	10	10
Networking and communication	3	7	10
Operating systems	4	11	15
Platform-based development	0	0	0
Parallel and distributed computing	5	10	15
Programming languages	8	20	28
Software development fundamentals	43	0	43
Software engineering	6	22	28
Systems fundamentals	18	9	27
Social issues and professional practice	11	5	16
Total core hours	**165**	**143**	**308**

Totals

All Tier1 + All Tier2	308 (8 courses)
All Tier1 + 90% of Tier2	294 (7 courses)
All Tier1 + 80% of Tier2	280 (7 courses)

Computer science degree flowchart



Computer Science - Computer Science (SCS2) Degree Flowchart 123 Credits Effective Fall 2015

Year 1

Fall

First Year Only

CS1000 (1) Explorations in Computing — f

MA1031(Concurrently)/ 1032(Concurrently)

CS1121 (3)/ CS1131*[a]* (5) Intro Prog. I — f,s,su

MA1031/1032

MA1160 (4) MA1161 (5) Calculus I — f,s,su

Spring

CS1122 (3) Intro Prog. II — f,s,su

MA2160 (4) Calculus II — f,s,su

First Year Only

UN1015 (3) Composition — f,s,su

UN1025 Global Issues or 3K-level Language — f,s,su

Year 2

Fall

MA 1160/1161

CS2311 (3) Discret Structures — f,s,su

CS1122/1131

CS2321 (3) Data Structures — f,s,su

MA1031/1032

MA2720 (4)/ MA3710 (3) Statistics — f,s,su

MA/Lab Science Elective (3/4) *[d]*

Spring

CS1122/1131

CS1141 (2) C for JAVA prog. — f,s,su

CS3425 (3) Databases — f,s

MA1160/1161

MA2330 (3) Linear Algebra — f,s

Technical Elective (3) *[f]*

Year 3

CS1122/ CS1131 → CS3421 (4) Comp Org — f,s

CS2311, CS2321 → CS3141 (3) Team Software Proj — f,s

CS2311 → CS3311 (3) Formal Models of Comp — f,s,su

CS2311,2321 CS1141 → CS3331 (3) Concurrent Computing — f,s

CS1141, CS3421 → CS3411 (3) Systems Programming — f,s,su

UN1015,UN1025

SS Elective (3) Dept. Required *[b]*

UN1015,UN1025

HU3120 (3) Sci and Tech Communication — f,s,su

Year 4

CS3141 → CS3000 (2) Ethical and Social Aspects of Comp. — f,s

CS1141, CS2321, CS3311,CS3421 → CS4121 (3) Programming Languages — f,s

CS2311, CS2321 → CS4321 (3) Intro to Algorithms — f,s

Check Pre-reqs → CS4xxx (3) *[c]* CS Elective — check schedule

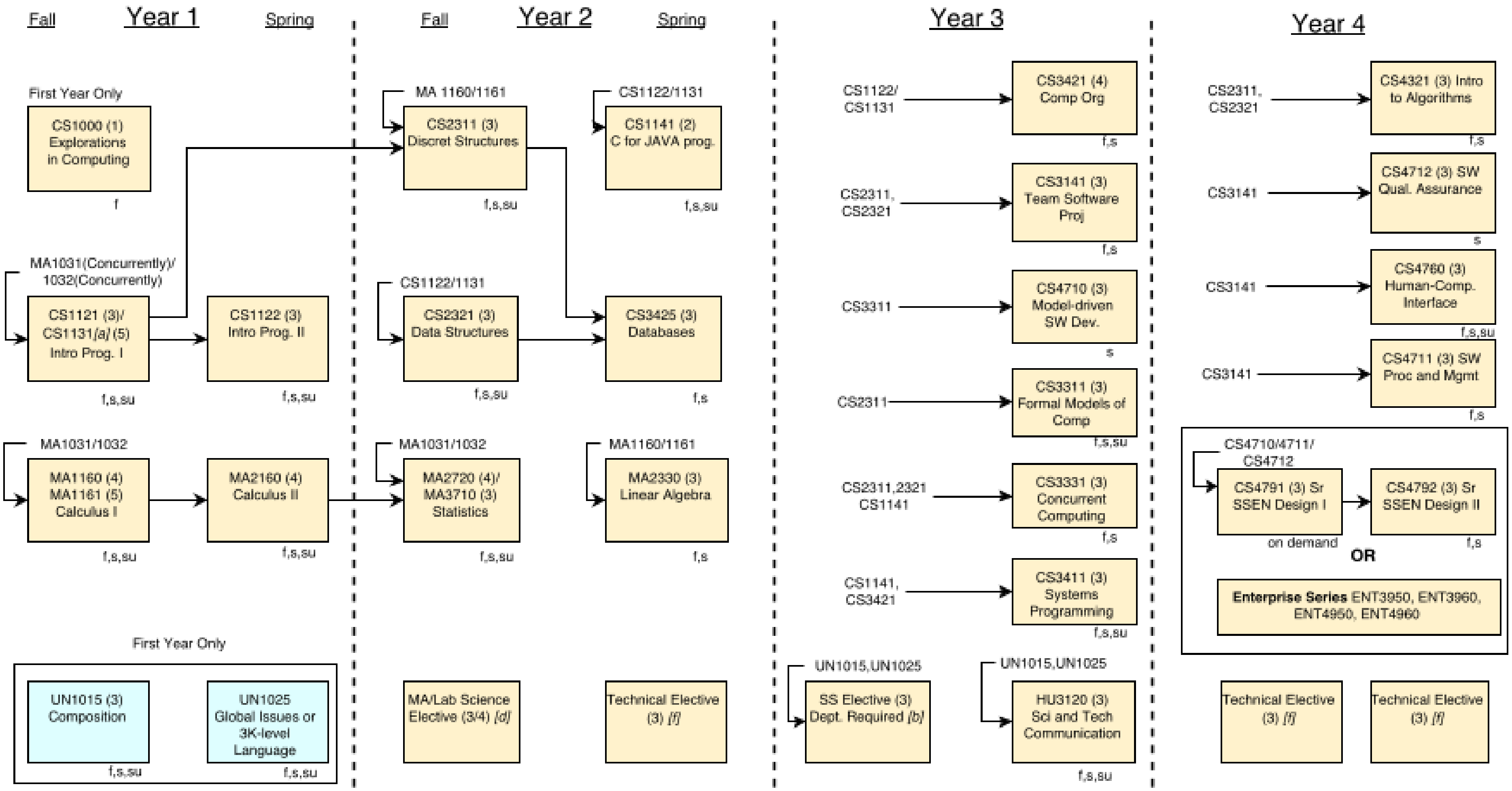
Check Pre-reqs → CS4yyy (3) *[c]* CS Elective — check schedule

Technical Elective (3) *[f]*

Technical Elective (3) *[f]*

Software engineering degree flowchart

Software Engineering (SSEN) Degree Flowchart 127 Credits Effective Fall 2015



Year 1

Fall

First Year Only

CS1000 (1) Explorations in Computing
f

MA1031(Concurrently)/ 1032(Concurrently)

CS1121 (3)/ CS1131*[a]* (5) Intro Prog. I
f,s,su

MA1031/1032

MA1160 (4) MA1161 (5) Calculus I
f,s,su

Spring

CS1122 (3) Intro Prog. II
f,s,su

MA2160 (4) Calculus II
f,s,su

First Year Only

UN1015 (3) Composition
f,s,su

UN1025 Global Issues or 3K-level Language
f,s,su

Year 2

Fall

MA 1160/1161

CS2311 (3) Discret Structures
f,s,su

CS1122/1131

CS2321 (3) Data Structures
f,s,su

MA1031/1032

MA2720 (4)/ MA3710 (3) Statistics
f,s,su

MA/Lab Science Elective (3/4) *[d]*

Spring

CS1122/1131

CS1141 (2) C for JAVA prog.
f,s,su

CS3425 (3) Databases
f,s

MA1160/1161

MA2330 (3) Linear Algebra
f,s

Technical Elective (3) *[f]*

Year 3

CS1122/ CS1131 → CS3421 (4) Comp Org
f,s

CS2311, CS2321 → CS3141 (3) Team Software Proj
f,s

CS3311 → CS4710 (3) Model-driven SW Dev.
s

CS2311 → CS3311 (3) Formal Models of Comp
f,s,su

CS2311,2321 CS1141 → CS3331 (3) Concurrent Computing
f,s

CS1141, CS3421 → CS3411 (3) Systems Programming
f,s,su

UN1015,UN1025

SS Elective (3) Dept. Required *[b]*

UN1015,UN1025

HU3120 (3) Sci and Tech Communication
f,s,su

Year 4

CS2311, CS2321 → CS4321 (3) Intro to Algorithms
f,s

CS3141 → CS4712 (3) SW Qual. Assurance
s

CS3141 → CS4760 (3) Human-Comp. Interface
f,s,su

CS3141 → CS4711 (3) SW Proc and Mgmt
f,s

CS4710/4711/ CS4712

CS4791 (3) Sr SSEN Design I
on demand

CS4792 (3) Sr SSEN Design II
f,s

OR

Enterprise Series ENT3950, ENT3960, ENT4950, ENT4960

Technical Elective (3) *[f]*

Technical Elective (3) *[f]*

Software Development Fundamentals (43 hours)

- Reading and writing programs in multiple programming languages
- Utilize modern development and testing tools
- Focuses on the entire software development process as well
- Includes:
    - Algorithms and design
    - Data structures

Discrete structures (41 hours)

- Foundational material: supports other areas
- Ability to create and understand a proof
  formal specification, verification, databases, cryptography
- Graph theory
  used in networks, operating systems, and compilers
- Logic, counting, discrete probability

Software engineering (28 hours)

- Software engineering is the discipline concerned with the application of theory, knowledge, and practice to effectively build reliable software systems that satisfy the requirements of customers and users
- Producing software systems: professionalism, quality, schedule, and cost are critical
- A wide variety of software engineering practices have been developed
- Consider trade-offs when selecting and applying different practices

Algorithms and complexity (28 hours)

- Good algorithm design is crucial for the performance of all software systems
- There are a range of algorithms that address an important set of well-defined problems
- Recognize their strengths and weaknesses, and their suitability in particular contexts

Programming languages (28 hours)

- Programming languages are the medium through which programmers precisely describe concepts, formulate algorithms, and reason about solutions
- Making informed design choices by understanding the languages supporting multiple complementary approaches
- Basic knowledge of programming language translation

Systems fundamentals (27 hours)

- The underlying hardware and software infrastructure upon which applications are constructed is collectively described by the term “computer systems”
- Broadly spans
  - Operating systems
  - Parallel and distributed systems
  - Communication networks
  - Computer architecture

Architecture and organization (16 hours)

- Understand the hardware environment upon which all computing is based, and the interface it provides to higher software layers
- Develop programs that can achieve high performance through a programmer's awareness of parallelism and latency
- Select a system use through an understanding of the trade-off among various components, such as CPU clock speed, cycles per instruction, memory size, and average memory access time.

Social issues and professional practice (16 hours)

- In addition to the technical issues in computing students must be exposed to the larger societal context of computing
- Developing an understanding of the relevant social, ethical, legal and professional issues
- Anticipate the impact of introducing a given product into a given environment
  - Enhance or degrade the quality of life
  - Impact upon individuals, groups, and institutions?
- Legal rights of software and hardware vendors and users, ethical values

Operating systems (15 hours)

- An operating system (O/S)
  - Defines an abstraction of hardware
  - Manages resource sharing among the computer's users
- Basic topics taught
  - Interface of an operating system to networks
  - Kernel and user modes
  - Approaches to O/S design and implementation

Parallel and distributed computing (15 hours)

- Was a largely elective topic before multi-core processors and distributed data centers
- Logically simultaneous execution of multiple processes whose operations have the potential to interleave in complex ways
- Models of communication and coordination among processes
- Security and fault tolerance in distributed systems

Summary

- Information about me (done)
- Tips to connect with faculty (done)
- Course information (done)
- Computer Science curricula (done)

- Reliability
- Correctness
- Performance
- Abstraction
- Layers
- Trade-offs
- Representation
- Algorithms