LUC CS Course Handbook

Release 1.0

CS Department

Feb 20, 2017

CONTENTS

Welcome to the Course Catalog for the Computer Science Department of Loyola University Chicago!

Below you can find information on all of our Course Offerings and Scedules as well as our Undergraduate and Graduate Degree Programs.

- *Schedules*
- *Undergraduate Degree Programs*
- *Undergraduate Courses*
- *Graduate Degree Programs*
- *Graduate Courses*
- *Downloading*

If you want another format of this document (html, pdf, ebook), see download_other_formats.

ONE

SCHEDULES

1.1 Fall 2017 Schedule

Updated 02/16/2017 17:33:59

The following courses will (tentatively) be held during the Fall 2017 semester.

For open/full status and latest changes, see LOCUS.

In case of conflict, information on LOCUS should be considered authoritative.

See Textbook Information.

Section titles lines link to the course description page, except for some labeled special topics courses related to an existing course.

The 4-digit number in parentheses after the section is the Locus registration code.

Be sure to look at the section's notes or Locus for an 8-week courses with more than one schedule line: Friday line(s) are likely to be isolated makeup days, not every week.

You can skip down to *Graduate Courses*.

View Campus Specific Courses below :

lakeShorefall

waterTowerFall

onlineFall

1.1.1 Undergraduate Courses

COMP 111: History of Computing

> Section 01W (4091) Credits: 3; Blended; Lecture
> Instructor: George Thiruvathukal
> Place TBA (Lake Shore) Monday, Wednesday, Friday 12:35PM-01:25

> **Notes:** *This is a writing intensive course.*

> This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 122: Introduction to Digital Music

> Section 001 (4796) Credits: 3; In person; Lecture
> Instructor: David Wetzel

Mundelein Center:1017 (Lake Shore) Monday, Wednesday, Friday 12:35PM-01:25

COMP 125: Visual Information Processing

Section 001 (3245) Credits: 3; In person; Laboratory

Instructor: David Wetzel

Place TBA (Lake Shore) Monday, Wednesday, Friday 10:25AM-11:15

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing

Section 001 (3246) Credits: 3; In person; Lecture

Instructor: Catherine Putonti

Place TBA (Lake Shore) Tuesday, Thursday 08:30AM-09:45

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing

Section 002 (3247) Credits: 3; In person; Lecture

Instructor: Elliott Lawrence Post

Place TBA (Lake Shore) Monday, Wednesday 04:15PM-05:30

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing

Section 003 (3248) Credits: 3; In person; Lecture

Instructor: Channah Naiman

Place TBA (Lake Shore) Monday, Wednesday, Friday 11:30AM-12:20

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 163: Discrete Structures

Section 001 (3249) Credits: 3; In person; Lecture

Instructor: Curtis D Tuckey

Place TBA (Lake Shore) Monday, Wednesday 02:45PM-04:00

Notes: This course is primarily intended to serve certain majors and minors. Students wishing to satisfy Core requirements in the Quantitative Analysis knowledge area are encouraged to enroll in COMP 125 or COMP 150 instead. This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 163: Discrete Structures

Section 002 (3514) Credits: 3; In person; Lecture

Instructor: Curtis D Tuckey

Place TBA (Lake Shore) Monday, Wednesday 04:15PM-05:30

Notes: This course is primarily intended to serve certain majors and minors. Students wishing to satisfy Core requirements in the Quantitative Analysis knowledge area are encouraged to enroll in COMP 125 or COMP 150 instead. This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 163: Discrete Structures

Section 003 (5993) Credits: 3; In person; Lecture

Instructor: Chandra N Sekharan

Place TBA (Lake Shore) Tuesday, Thursday 01:00PM-02:15

Notes: This course is primarily intended to serve certain majors and minors. Students wishing to satisfy Core requirements in the Quantitative Analysis knowledge area are encouraged to enroll in COMP 125 or COMP 150 instead. This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 001 (3250) Credits: 3; Blended; Lecture

Instructor: Robert Yacobellis

Place TBA (Lake Shore) Tuesday, Thursday 10:00AM-11:15

Notes: This course is restricted to undergraduate students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 002 (3251) Credits: 3; In person; Lecture

Instructor: Dmitriy Dligach

Place TBA (Lake Shore) Tuesday, Thursday 02:30PM-04:05

Notes: This course is restricted to undergraduate students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 003 (3626) Credits: 3; In person; Lecture

Instructor: William Honig

Place TBA (Lake Shore) Monday, Wednesday 09:20AM-11:15

Notes: This course is restricted to undergraduate students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 004/04L (3860) Credits: 3; In person; Lecture/Lab

Instructor: William Honig

Place TBA (Lake Shore) Monday, Wednesday, Friday 11:30AM-12:20

Place TBA (Lake Shore) Friday 10:25AM-11:15 (lab)

Notes: This course is restricted to undergraduate students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

Students enrolled in COMP 170-004 (Lecture) will be automatically enrolled in COMP 170-04L (Lab).

COMP 170: Introduction to Object-Oriented Programming [**Term: Eight Week 1 - Alternate**]

Section 400 (6220) Credits: 3; In person; Lecture

Instructor: Andrew N Harrington

Place TBA (Water Tower) Monday 04:15PM-08:15

Place TBA (Water Tower) Thursday 04:15PM-06:30

Place TBA (Water Tower) Friday 04:15PM-08:15 - Check week(s)

Notes: Foundations of Computer Science I. Eight Week-First Session.

This section is restricted to students with undergraduate degrees. Department Consent required, and then a Computer Science Department staff member will enroll you.

COMP 170-400 meets on Mondays, 4:15 pm - 8:15 pm, for the first eight weeks of the Semester, replacing holiday/vacation Mondays with Fridays. So the Monday/Friday class schedule is: Mon, Aug 28; Fri, Sept 8; Mon, Sept 11; Mon, Sept 18; Mon, Sept 25; Mon, Oct 2; Fri, Oct 13; and Mon, Oct 16.

Labs meet on consecutive Thursdays, 4:15 pm - 6:30 pm: Thurs, Aug 31 through Thurs, Oct 19.

COMP 170: Introduction to Object-Oriented Programming [**Term: Eight Week - First**]

Section 700N (5217) Credits: 3; Online; Lecture

Instructor: Safoora Fatima

Online Tuesday 07:30PM-09:00

Notes: COMP 170-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 170-700N is an online section. Required class meetings will be held Tuesdays 7:30-9PM CST.

COMP 215: Object-Oriented Programming with Mathematics

Section 001 (3274) Credits: 3; In person; Lecture

Instructor: Aaron Lauve

Place TBA (Lake Shore) Monday, Wednesday, Friday 02:45PM-03:35

Notes: COMP 215 is crosslisted with MATH 215. Register for MATH 215.

COMP 250: Introduction to Scientific and Technical Communication

Section 01W (3425) Credits: 3; In person; Lecture

Instructor: Roxanne Schwab

Place TBA (Lake Shore) Monday, Wednesday 02:45PM-04:00

Notes: *This is a writing intensive course.* This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 251: Introduction to Database Systems

Section 001 (3361) Credits: 3; In person; Lecture

Instructor: Guy Bevente

Place TBA (Lake Shore) Monday 07:00PM-09:30

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 251: Introduction to Database Systems [**Term: Eight Week - Second**]

Section 700N (5222) Credits: 3; Online; Lecture

Instructor: Safoora Fatima

Online Tuesday 07:30PM-09:00

Notes: COMP 251-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 251-700N is an online section. Required synchronous sessions will be held Tuesdays 7:30-9PM CST

COMP 264: Introduction to Computer Systems

Section 001 (3627) Credits: 3; In person; Lecture
Instructor: Ronald I Greenberg
Place TBA (Lake Shore) Monday, Wednesday, Friday 09:20AM-10:10

COMP 271: Data Structures

Section 001 (3252) Credits: 3; In person; Lecture
Instructor: Mark Albert
Place TBA (Lake Shore) Tuesday, Thursday 02:30PM-04:05

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 271: Data Structures

Section 002/02L (3628) Credits: 3; Blended; Lecture/Lab
Instructor: Konstantin Laufer
Place TBA (Lake Shore) Monday, Wednesday, Friday 08:15AM-09:05
Place TBA (Lake Shore) Friday 09:20AM-10:10 (lab)

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

Students enrolled in COMP 271-002 (Lecture) will be automatically enrolled in COMP 271-02L (Lab).

COMP 271: Data Structures [**Term: Eight Week - Second**]

Section 400 (6107) Credits: 3; In person; Lecture
Instructor: Peter L Dordal
Place TBA (Water Tower) Monday 04:15PM-08:15
Place TBA (Water Tower) Thursday 04:15PM-06:30

COMP 305: Database Administration

Section 001 (5994) Credits: 3; Online; Lecture
Instructor: Staff
Online Times: TBA

Notes: This is an online class. Combined with COMP 488-305.

COMP 312: Free/Open Source Computing

Section 01E (4045) Credits: 3; Blended; Lecture
Instructor: Michael Lewis
Place TBA (Lake Shore) Thursday 07:00PM-09:30

Notes: This is a hybrid class. Combined with COMP 412-001.

This class satisfies the Engaged Learning requirement in the Undergraduate Research category.

COMP 313: Intermediate Object-Oriented Development

Section 001 (3740) Credits: 3; In person; Lecture
Instructor: Robert Yacobellis
Place TBA (Lake Shore) Tuesday, Thursday 01:00PM-02:15

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 314 (Description: *COMP 314/315: Problem Solving Strategies***)**

Section 001 (4950) Credits: 1; In person; Seminar

Instructor: Andrew N Harrington
Place TBA (Lake Shore) Times: TBA

Notes: Organizational meeting on the first day of classes: Monday, August 28th, 3:00 pm - 4:00 pm, to arrange upcoming practice times and place. Contact Dr. Andrew Harrington (aharrin@luc.edu) beforehand if you cannot attend, or if you have any questions.

COMP 315 (Description: *COMP 314/315: Problem Solving Strategies*)

Section 001 (4803) Credits: 2; In person; Seminar
Instructor: Andrew N Harrington
Place TBA (Lake Shore) Times: TBA

Notes: Organizational meeting on the first day of classes: Monday, August 28th, 3:00 pm - 4:00 pm, to arrange upcoming practice times and place. Contact Dr. Andrew Harrington (aharrin@luc.edu) beforehand if you cannot attend, or if you have any questions.

COMP 317: Social, Legal, and Ethical Issues in Computing

Section 001 (3254) Credits: 3; Online; Lecture
Instructor: Matthew Paul Butcher
Online Tuesday 07:00PM-09:30

Notes: This is an online class. All lectures will be pre-recorded. Students are asked to attend smaller-group online interactive discussions at regular intervals during the semester, with possible times chosen to fit different groups' schedules.

This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 322: Software Development for Wireless/Mobile Devices

Section 001 (4804) Credits: 3; In person; Lecture
Instructor: Nicholas J Hayward
Place TBA (Lake Shore) Friday 02:45PM-05:15

Notes: Combined with COMP 422-001

COMP 324: Client-Side Web Development

Section 001 (4820) Credits: 3; In person; Lecture
Instructor: Nicholas J Hayward
Place TBA (Water Tower) Monday 04:15PM-06:45

Notes: Combined with COMP 424-001

COMP 330: Software Engineering

Section 001 (6005) Credits: 3; Blended; Lecture
Instructor: George Thiruvathukal
Place TBA (Lake Shore) Monday, Wednesday, Friday 01:40PM-02:30

COMP 331: Cryptography

Section 001 (5846) Credits: 3; In person; Lecture
Instructor: Aaron Greicius
Place TBA (Lake Shore) Monday, Wednesday, Friday 10:25AM-11:15

Notes: COMP 331 is cross listed with MATH 331. Please register for MATH 331.

COMP 336: Markup Languages

Section 001 (4821) Credits: 3; In person; Lecture
Instructor: Nicholas J Hayward
Place TBA (Water Tower) Wednesday 04:15PM-06:45

Notes: Combined with COMP 436-001

COMP 343: Introduction to Computer Networks

Section 001 (4823) Credits: 3; In person; Lecture
Instructor: Peter L Dordal
Place TBA (Water Tower) Tuesday 04:15PM-06:45

Notes: Combined with COMP 443-001

COMP 343: Introduction to Computer Networks

Section 002 (4826) Credits: 3; Online; Lecture
Instructor: Peter L Dordal
Online Times: TBA

Notes: This is an online class that includes synchronous and asynchronous interaction among students and Instructor. Synchronous discussion sessions will be held Mondays and Tuesdays at 2:30 pm, and may vary in length from 30 minutes to one hour. Participation in synchronous sessions is strongly recommended. Combined with COMP 443-002.

COMP 344: Web Services Programming

Section 001 (4827) Credits: 3; In person; Lecture
Instructor: Berhane Zewdie
Place TBA (Water Tower) Tuesday 07:00PM-09:30

Notes: Combined with COMP 433-001

COMP 346: Introduction to Telecommunications

Section 001 (6006) Credits: 3; Blended; Lecture
Instructor: Staff
Place TBA (Water Tower) Wednesday 07:00PM-09:30

Notes: Combined with COMP 446-001

COMP 347: Intrusion Detection and Computer Security

Section 001 (4830) Credits: 3; In person; Lecture
Instructor: Corby Schmitz
Place TBA (Water Tower) Friday 05:45PM-08:15

Notes: Combined with COMP 447-001

COMP 347: Intrusion Detection and Computer Security

Section 002 (4831) Credits: 3; Online; Lecture
Instructor: Corby Schmitz
Online Times: TBA

Notes: This is an online class. The classroom session will be broadcast live on Friday evenings via Adobe-Connect, allowing online student interaction. Sessions will also be recorded and made available. Students may participate synchronously or asynchronously at their discretion. Combined with COMP 447-002.

COMP 363: Design and Analysis of Computer Algorithms

Section 001 (3264) Credits: 3; In person; Lecture

Instructor: Chandra N Sekharan
Place TBA (Lake Shore) Tuesday, Thursday 11:30AM-12:45

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 364: High-Performance Computing

Section 001 (4832) Credits: 3; Online; Lecture
Instructor: Christopher Stone
Online Wednesday 07:00PM-09:30

Notes: This is an online, synchronous class. Synchronous meeting time: Wednesday, 7:00 pm - 9:30 pm. Combined with COMP 464-001.

COMP 374: Introduction to Operating Systems

Section 001 (4834) Credits: 3; In person; Lecture
Instructor: Sarah Kaylor
Place TBA (Lake Shore) Thursday 07:00PM-09:30

Notes: Combined with COMP 410-001

COMP 377: IT Project Management

Section 001 (6007) Credits: 3; In person; Lecture
Instructor: Conrad Weisert
Place TBA (Water Tower) Monday 07:00PM-09:30

Notes: Combined with COMP 477-001

COMP 379: Machine Learning

Section 001 (4835) Credits: 3; In person; Lecture
Instructor: Dmitriy Dligach
Place TBA (Lake Shore) Tuesday, Thursday 08:30AM-09:45

Notes: Combined with COMP 488-379

COMP 381: Bioinformatics

Section 002 (4101) Credits: 3; In person; Lecture
Instructor: Heather E. Wheeler
Place TBA (Lake Shore) Monday, Wednesday 02:45PM-04:00

Notes: COMP 381-002 is combined with BIOL 388-001. Register for BIOL 388-001 (2013). Also, combined with COMP 488-381 and BIOL 488-001.

COMP 384: Metagenomics

Section 01E (6018) Credits: 3; In person; Lecture
Instructor: Staff
Place TBA (Lake Shore) Tuesday, Thursday 08:30AM-09:45

Notes: This class satisfies the Engaged Learning requirement in the Undergraduate Research category. Instructor Consent Required.

Combined with COMP 488-384 and BIOL 392-001.

COMP 386: Computational Neuroscience

Section 001 (6008) Credits: 3; In person; Lecture
Instructor: Mark Albert

Place TBA (Lake Shore) Tuesday, Thursday 10:00AM-11:15

Notes: Combined with COMP 488-386

COMP 388: Topics in Computer Science: **DevOps: MethodsToolsPractices**

Section 001 (6009) Credits: 3; In person; Lecture
Instructor: Maria Del Carmen Saenz
Place TBA (Water Tower) Friday 05:45PM-08:15

Notes: DevOps: Methods, Tools & Practices

Combined with COMP 488-001

COMP 388 Topic: RapidAppDevMethodology(.NET)

Section 325 (6010) Credits: 3; Blended; Lecture
Instructor: Karim Kabani
Place TBA (Lake Shore) Saturday 10:00AM-12:30
Description similar to: comp325

Notes: Rapid App Dev Methodology (.NET)

Combined with COMP 488-325

COMP 388 Topic: Data Warehousing

Section 796 (6011) Credits: 3; In person; Lecture
Instructor: Staff
Place TBA (Water Tower) Thursday 07:00PM-09:30
Description similar to: comp796

Notes: Data Warehousing

Combined with COMP 488-796

COMP 390: Broadening Participation in STEM (Computing, Math & Science)

Section 01E (3742) Credits: 3; Online; Lecture
Instructor: Ronald I Greenberg
Online Times: TBA

Notes: Broadening Participation in STEM (Computing, Mathematics, and Science).

This class is online and fully asynchronous, but students must complete service learning activities in-person at a site of their choosing to be approved by the instructor in accord with the course design.

This class satisfies the Engaged Learning requirement in the Service Learning category.

COMP 391: Internship in Computer Science

Section 01E (2173) Credits: 1 - 6; In person; Field Studies
Instructor: Ronald I Greenberg, Robert Yacobellis
Place TBA (Lake Shore) Times: TBA

Notes: This class satisfies the Engaged Learning requirement in the Internship category. Department Consent is required, and then a Computer Science Department staff member will enroll you.

COMP 391: Internship in Computer Science

Section 02E (6120) Credits: 1 - 6; Online; Field Studies
Instructor: Ronald I Greenberg, Robert Yacobellis
Online Times: TBA

Notes: This is an online class. This class satisfies the Engaged Learning requirement in the Internship category. Department Consent is required, and then a Computer Science Department staff member will enroll you.

COMP 398: Independent Study **1-6 credits** You cannot register yourself for an independent study course! You must find a faculty member who agrees to supervisor the work that you outline and schedule together. This *supervisor arranges to get you registered.* Possible supervisors are:

COMP 399: Computer Science Seminar

> Section 001 (6012) Credits: 1; In person; Lecture
> Instructor: Mark Albert
> Place TBA (Lake Shore) Thursday 04:15PM-05:30

1.1.2 Graduate Courses

COMP 410: Operating Systems

> Section 001 (4837) Credits: 3; In person; Lecture
> Instructor: Sarah Kaylor
> Place TBA (Lake Shore) Thursday 07:00PM-09:30

> **Notes:** Combined with COMP 374-001

COMP 412: Free/Open Source Computing

> Section 001 (4052) Credits: 3; Blended; Lecture
> Instructor: Michael Lewis
> Place TBA (Lake Shore) Thursday 07:00PM-09:30

> **Notes:** Combined with COMP 312-01E

COMP 413: Intermediate Object-Oriented Development

> Section 001 (3741) Credits: 3; In person; Lecture
> Instructor: Robert Yacobellis
> Place TBA (Lake Shore) Thursday 04:15PM-06:45

COMP 417: Social, Legal, and Ethical Issues in Computing

> Section 001 (3255) Credits: 3; In person; Lecture
> Instructor: Roxanne Schwab
> Place TBA (Lake Shore) Wednesday 04:15PM-06:45

COMP 422: Software Development for Wireless/Mobile Devices

> Section 001 (4843) Credits: 3; In person; Lecture
> Instructor: Nicholas J Hayward
> Place TBA (Lake Shore) Friday 02:45PM-05:15

> **Notes:** Combined with COMP 322-001

COMP 424: Client-Side Web Design

> Section 001 (4844) Credits: 3; In person; Lecture
> Instructor: Nicholas J Hayward
> Place TBA (Water Tower) Monday 04:15PM-06:45

> **Notes:** Combined with COMP 324-001

COMP 431: Cryptography

Section 001 (5847) Credits: 3; In person; Lecture

Instructor: Aaron Greicius

Place TBA (Lake Shore) Monday, Wednesday, Friday 10:25AM-11:15

Notes: COMP 431 is cross listed with MATH 431. Please register for MATH 431.

COMP 433: Web Services Programming

Section 001 (4845) Credits: 3; In person; Lecture

Instructor: Berhane Zewdie

Place TBA (Water Tower) Tuesday 07:00PM-09:30

Notes: Combined with COMP 344-001

COMP 436: Markup Languages

Section 001 (4846) Credits: 3; In person; Lecture

Instructor: Nicholas J Hayward

Place TBA (Water Tower) Wednesday 04:15PM-06:45

Notes: Prerequisite for students in the MA in Digital Humanities program: DIGH 402 taken in spring 2016 or previously, or COMP 271. Combined with COMP 336-001

COMP 443: Computer Networks

Section 001 (4847) Credits: 3; In person; Lecture

Instructor: Peter L Dordal

Place TBA (Water Tower) Tuesday 04:15PM-06:45

Notes: Combined with COMP 343-001

COMP 443: Computer Networks

Section 002 (4848) Credits: 3; Online; Lecture

Instructor: Peter L Dordal

Online Times: TBA

Notes: This is an online class that includes synchronous and asynchronous interaction among students and Instructor. Synchronous discussion sessions will be held Mondays and Tuesdays at 2:30 pm, and may vary in length from 30 minutes to one hour. Participation in synchronous sessions is strongly recommended. Combined with COMP 343-002.

COMP 446: Telecommunications

Section 001 (6064) Credits: 3; Blended; Lecture

Instructor: Staff

Place TBA (Water Tower) Wednesday 07:00PM-09:30

Notes: Combined with COMP 346-001

COMP 447: Intrusion Detection

Section 001 (4849) Credits: 3; In person; Lecture

Instructor: Corby Schmitz

Place TBA (Water Tower) Friday 05:45PM-08:15

Notes: Combined with COMP 347-001

COMP 447: Intrusion Detection

Section 002 (4850) Credits: 3; Online; Lecture

Instructor: Corby Schmitz

Online Times: TBA

Notes: This is an online class. The classroom session will be broadcast live on Friday evenings via Adobe-Connect, allowing online student interaction. Sessions will also be recorded and made available. Students may participate synchronously or asynchronously at their discretion. Combined with COMP 347-002.

COMP 453: Database Programming

Section 001 (3267) Credits: 3; In person; Lecture
Instructor: Channah Naiman
Place TBA (Water Tower) Tuesday 04:15PM-06:45

Notes: This is a on-campus class, with regular attendance required in the classroom. The course will use a flipped" approach for the second half of the course." Some lectures may be online.

COMP 453: Database Programming

Section 002 (4053) Credits: 3; Online; Lecture
Instructor: Channah Naiman
Online Times: TBA

Notes: This is an online, asynchronous class. Students may be asked to attend smaller-group online interactive discussions during the semester, with possible times chosen to fit different groups' schedules. Exams may be synchronous and in-person.

COMP 464: High-Performance Computing

Section 001 (4852) Credits: 3; Online; Lecture
Instructor: Christopher Stone
Online Wednesday 07:00PM-09:30

Notes: Combined with COMP 364-001

COMP 477: IT Project Management

Section 001 (6065) Credits: 3; In person; Lecture
Instructor: Conrad Weisert
Place TBA (Water Tower) Monday 07:00PM-09:30

Notes: Combined with COMP 377-001

COMP 488: Topics in Computer Science: **DevOps: MethodsToolsPractices**

Section 001 (6066) Credits: 3; In person; Lecture
Instructor: Maria Del Carmen Saenz
Place TBA (Water Tower) Friday 05:45PM-08:15

Notes: DevOps: Methods, Tools & Practices

Combined with COMP 388-001

COMP 488 Topic: Database Administration

Section 305 (6067) Credits: 3; Online; Lecture
Instructor: Staff
Online Times: TBA
Description similar to: *COMP 305: Database Administration*

Notes: Database Administration

Combined with COMP 305-001

COMP 488 Topic: RapidAppl Dev Metholodgy(.NET)

Section 325 (6068) Credits: 3; Blended; Lecture
Instructor: Karim Kabani
Place TBA (Lake Shore) Saturday 10:00AM-12:30
Description similar to: comp325

Notes: Rapid App Dev Methodology (.NET)

Combined with COMP 388-325

COMP 488 Topic: Machine Learning

Section 379 (6069) Credits: 3; In person; Lecture
Instructor: Dmitriy Dligach
Place TBA (Lake Shore) Tuesday, Thursday 08:30AM-09:45
Description similar to: *COMP 379: Machine Learning*

Notes: Machine Learning

Combined with COMP 379-001

COMP 488 Topic: Bioinformatics

Section 381 (4856) Credits: 3; In person; Lecture
Instructor: Heather E. Wheeler
Place TBA (Lake Shore) Monday, Wednesday 02:45PM-04:00
Description similar to: *COMP 381: Bioinformatics*

Notes: Bioinformatics

Combined with COMP 381-002, BIOL 388-001, and BIOL 488-001

COMP 488 Topic: Metagenomics

Section 384 (6019) Credits: 3; In person; Lecture
Instructor: Staff
Place TBA (Lake Shore) Tuesday, Thursday 08:30AM-09:45
Description similar to: *COMP 384: Metagenomics*

Notes: Metagenomics

Instructor Consent Required

Combined with COMP 384-001 and BIOL 392-001

COMP 488 Topic: Computational Neurosci

Section 386 (6070) Credits: 3; In person; Lecture
Instructor: Mark Albert
Place TBA (Lake Shore) Tuesday, Thursday 10:00AM-11:15
Description similar to: *COMP 386: Computational Neuroscience*

Notes: Computational Neuroscience

Combined with COMP 386-001

COMP 488 Topic: Data Warehouse & Mining

Section 796 (6071) Credits: 3; In person; Lecture
Instructor: Staff
Place TBA (Water Tower) Thursday 07:00PM-09:30
Description similar to: comp796

Notes: Data Warehousing

Combined with COMP 388-796

COMP 490: Independent Project **1-6 credits** You cannot register yourself for an independent study course! You must find a faculty member who agrees to supervisor the work that you outline and schedule together. This *supervisor arranges to get you registered.* Possible supervisors are:

COMP 499: Internship

Section 001 (2182) Credits: 1 - 6; In person; Independent Study
Instructor: Andrew N Harrington, Channah Naiman
Place TBA (Water Tower) Times: TBA

Notes: This course involves an internship experience. Department Consent required, and then a Computer Science Department staff member will enroll you.

COMP 499: Internship

Section 002 (6121) Credits: 1 - 6; Online; Independent Study
Instructor: Andrew N Harrington, Channah Naiman
Online Times: TBA

Notes: This is an online class. This course involves an internship experience. Department Consent required, and then a Computer Science Department staff member will enroll you.

COMP 605: Master of Science Study

Section 001 (3077) Credits: 0; In person; FTC-Supervision
Instructor: Staff
Place TBA (Water Tower) Times: TBA

Notes: Department Consent required, and then a Computer Science Department staff member will enroll you.

1.2 Summer 2017 Schedule

The CAS has posted the course schedule for Summer 2017.

Note that Independent Study and Internships can be taken during any session (A, B, or C) with B being the most popular choice as grades are posted later.

The schedule can be found here: http://www.luc.edu/summer/courses/cas/ (scroll down the page to the Computer Science Schedule).

1.3 Spring 2017 Schedule

Updated 11/11/2016 13:49:55

The following courses will (tentatively) be held during the Spring 2017 semester.

For open/full status and latest changes, see LOCUS.

In case of conflict, information on LOCUS should be considered authoritative.

See Textbook Information.

Section titles lines link to the course description page, except for some labeled special topics courses related to an existing course.

The 4-digit number in parentheses after the section is the Locus registration code.

Be sure to look at the section's notes or Locus for an 8-week courses with more than one schedule line: Friday line(s) are likely to be isolated makeup days, not every week.

You can skip down to *Graduate Courses*.

View Campus Specific Courses below :

lakeShoreSpring

waterTowerSpring

onlineSpring

1.3.1 Undergraduate Courses

COMP 111: History of Computing

> Section 01W (4308) Credits: 3; Blended; Lecture
>
> Instructor: George Thiruvathukal
>
> Information Commons:216 (Lake Shore) Monday, Wednesday, Friday 11:30AM-12:20
>
> **Notes:** *This is a writing intensive course.* This course is blended/online (33%). MW meetings are on-campus, and F meetings are online and asynchronous. This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 125: Visual Information Processing

> Section 001 (2799) Credits: 3; In person; Laboratory
>
> Instructor: David Wetzel
>
> Crown Center:105 (Lake Shore) Tuesday, Thursday 01:00PM-02:15
>
> **Notes:** This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 125: Visual Information Processing

> Section 002 (2910) Credits: 3; In person; Laboratory
>
> Instructor: George Thiruvathukal
>
> Crown Center:105 (Lake Shore) Monday, Wednesday, Friday 10:25AM-11:15
>
> **Notes:** This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing

> Section 001 (2800) Credits: 3; In person; Lecture
>
> Instructor: Jason E Streeter
>
> Crown Center:105 (Lake Shore) Monday, Wednesday, Friday 11:30AM-12:20
>
> **Notes:** This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing

> Section 002 (2803) Credits: 3; In person; Lecture
>
> Instructor: Curtis D Tuckey
>
> Inst for Environment:111 (Lake Shore) Monday, Wednesday 04:15PM-05:30
>
> **Notes:** This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing

Section 003 (2804) Credits: 3; In person; Lecture

Instructor: Heather E. Wheeler

Crown Center:103 (Lake Shore) Monday, Wednesday 12:10PM-01:25

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 150: Introduction to Computing **[Term: Eight Week - Second]**

Section 700N (4778) Credits: 3; Online; Lecture

Instructor: Safoora Fatima

Online Monday 07:30PM-09:00

Notes: COMP 150-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 150-700N is an online section. Required synchronous sessions will be held Mondays 7:30-9PM CST.

COMP 163: Discrete Structures

Section 001 (3383) Credits: 3; Blended; Lecture

Instructor: Victoria Noquez

Cuneo Hall:324 (Lake Shore) Monday, Wednesday 02:45PM-04:00

Notes: This course is offered in a blended format, with about the final third of class sessions to be held online. Class sessions during the first two thirds of the term will meet on campus as scheduled. Details on schedule and format of the online sessions will be provided as part of the course syllabus at the beginning of the term.

This course is primarily intended to serve certain majors and minors. Students wishing to satisfy Core requirements in the Quantitative Analysis knowledge area are encouraged to enroll in COMP 125 or COMP 150 instead. This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 163: Discrete Structures

Section 002 (4232) Credits: 3; Blended; Lecture

Instructor: Victoria Noquez

Cuneo Hall:324 (Lake Shore) Monday, Wednesday 04:15PM-05:30

Notes: This course is offered in a blended format, with about the final third of class sessions to be held online. Class sessions during the first two thirds of the term will meet on campus as scheduled. Details on schedule and format of the online sessions will be provided as part of the course syllabus at the beginning of the term.

This course is primarily intended to serve certain majors and minors. Students wishing to satisfy Core requirements in the Quantitative Analysis knowledge area are encouraged to enroll in COMP 125 or COMP 150 instead. This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 001/01L (3213) Credits: 3; In person; Lecture/Lab

Instructor: Robert Yacobellis

Crown Center:103 (Lake Shore) Tuesday, Thursday 10:00AM-11:15

Crown Center:103 (Lake Shore) Tuesday, Thursday 11:30AM-11:55 (lab)

Notes: Restricted to Engineering Science students. Students from other schools must register for one of the other sections of this course. Students enrolled in COMP 170-001 (Lecture) will be automatically enrolled in COMP 170-01L (Lab). This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 002/02L (3214) Credits: 3; In person; Lecture/Lab

Instructor: Dmitriy Dligach

Crown Center:103 (Lake Shore) Monday, Wednesday 10:25AM-11:40

Crown Center:103 (Lake Shore) Friday 10:25AM-11:15 (lab)

Notes: This course is restricted to students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Students enrolled in COMP 170-002 (Lecture) will be automatically enrolled in COMP 170-02L (Lab). This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 003/03L (3216) Credits: 3; In person; Lecture/Lab

Instructor: William Honig

Crown Center:105 (Lake Shore) Tuesday, Thursday 11:30AM-12:45

Crown Center:105 (Lake Shore) Tuesday, Thursday 10:50AM-11:15 (lab)

Notes: This course is restricted to students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Students enrolled in COMP 170-003 (Lecture) will be automatically enrolled in COMP 170-03L (Lab). This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming

Section 004/04L (3375) Credits: 3; In person; Lecture/Lab

Instructor: Dmitriy Dligach

Cuneo Hall:111 (Lake Shore) Monday, Wednesday 12:10PM-01:25

Cuneo Hall:111 (Lake Shore) Friday 12:35PM-01:25 (lab)

Notes: This course is restricted to students in the College of Arts and Sciences. Students from other schools specifically wishing to enroll in COMP 170 should contact the Computer Science Department to request an enrollment override. Students enrolled in COMP 170-004 (Lecture) will be automatically enrolled in COMP 170-04L (Lab). This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 170: Introduction to Object-Oriented Programming **[Term: Eight Week 1 - Alternate]**

Section 400 (6052) Credits: 3; In person; Lecture

Instructor: Chandra N Sekharan

Corboy Law Center:0201 (Water Tower) Friday 04:15PM-08:15

Lewis Towers:415 (Water Tower) Wednesday 04:15PM-06:30

Lewis Towers:410 (Water Tower) Monday 04:15PM-08:15

Corboy Law Center:0201 (Water Tower) Friday 04:15PM-08:15 - Check week(s)

Notes: Foundations of Computer Science I. Eight Week-First Session.

This section is restricted to students with undergraduate degrees. Department Consent required, and then a Computer Science Department staff member will enroll you.

FIRST CLASS EARLY: Friday before other classes start: Friday, January 13! And then on for the next seven weeks, ending before vacation week. Classes meet from 4:15 pm - 8:15 pm on Fri, Jan 13, and Fri, Jan 20. And then, 4:15 pm- 8:15 pm for the next 6 Mondays: Jan 23, Jan 30, Feb 6, Feb 13, Feb 20, Feb 27. Labs meet on Wednesdays, 4:15 pm - 6:30 pm: Jan 18, Jan 25, Feb 1, Feb 8, Feb 15, Feb 22, Mar 1.

COMP 250: Introduction to Scientific and Technical Communication

Section 01W (2798) Credits: 3; In person; Lecture

Instructor: Roxanne Schwab

Crown Center:105 (Lake Shore) Monday, Wednesday 02:45PM-04:00

Notes: This is a writing intensive class. This class is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 251: Introduction to Database Systems **[Term: Eight Week - First]**

Section 700N (5970) Credits: 3; Online; Lecture

Instructor: Safoora Fatima

Online Monday 07:30PM-09:00

Notes: COMP 251-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 251-700N is an online section. Required synchronous sessions will be held Mondays 7:30-9PM CST.

COMP 264: Introduction to Computer Systems

Section 001 (2797) Credits: 3; In person; Lecture

Instructor: Ronald I Greenberg

Dumbach Hall:125 (Lake Shore) Monday, Wednesday, Friday 10:25AM-11:15

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 271: Data Structures

Section 001/01L (2796) Credits: 3; In person; Lecture/Lab

Instructor: Mark Albert

Crown Center:103 (Lake Shore) Tuesday, Thursday 01:00PM-02:15

Crown Center:103 (Lake Shore) Tuesday, Thursday 12:20PM-12:45 (lab)

Notes: Students enrolled in COMP 271-001 (Lecture) will be automatically enrolled in COMP 271-01L (Lab). This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 271: Data Structures

Section 002/02L (2805) Credits: 3; In person; Lecture/Lab

Instructor: Matthew William Bone

Crown Center:103 (Lake Shore) Wednesday 06:00PM-06:50

Crown Center:103 (Lake Shore) Wednesday 07:00PM-09:30 (lab)

Notes: Students enrolled in COMP 271-002 (Lecture) will be automatically enrolled in COMP 271-02L (Lab). This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 271: Data Structures **[Term: Eight Week - Second]**

Section 400 (6051) Credits: 3; In person; Lecture

Instructor: Peter L Dordal

Lewis Towers:410 (Water Tower) Monday 04:15PM-08:15

Lewis Towers:415 (Water Tower) Wednesday 04:15PM-06:30

Notes: Foundations of Computer Science II. Eight Week-Second Session.

This section is restricted to students with undergraduate degrees. Department Consent required, and then a Computer Science Department staff member will enroll you.

Eight weeks after spring break, including two meetings in finals week. Mondays, 4:15 pm -8:15 pm: March 13, March 20, March 27, April 3, April 10, April 17, April 24, May 1. Labs meet on Wednesdays, 4:15 pm - 6:30 pm: March 15, March 22, March 29, April 5, April 12, April 19, April 26, May 3.

COMP 271: Data Structures [**Term: Eight Week - First**]

Section 700N (5981) Credits: 3; Online; Lecture
Instructor: Elliott Lawrence Post
Online Thursday 06:00PM-09:00

Notes: COMP 271-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 271-700N is an online section. Required synchronous sessions will be held Thursdays 6-9PM CST.

COMP 300: Data Warehousing and Data Mining

Section 001 (5604) Credits: 3; In person; Lecture
Instructor: Channah Naiman
Corboy Law Center:0201 (Water Tower) Wednesday 04:15PM-06:45

Notes: Combined with COMP 488-301

COMP 300: Data Warehousing and Data Mining

Section 002 (5605) Credits: 3; Online; Lecture
Instructor: Channah Naiman
Online Times: TBA

Notes: This class is a totally online, asynchronous course. Exams may be synchronous and in person. Combined with COMP 488-302.

COMP 305: Database Administration

Section 001 (5606) Credits: 3; In person; Lecture
Instructor: Peter L Dordal
School of Communicat:010 (Water Tower) Tuesday 04:15PM-06:45

Notes: Combined with COMP 488-305

COMP 305: Database Administration [**Term: Eight Week - Second**]

Section 700N (6011) Credits: 3; Online; Lecture
Instructor: Sargon Hasso
Online Wednesday 06:00PM-09:00

Notes: COMP 305-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 305-700N is an online class. Required synchronous sessions will be held Wednesdays 6-9PM CST

COMP 312: Free/Open Source Computing

Section 01E (5607) Credits: 3; In person; Lecture
Instructor: Michael Lewis
Crown Center:105 (Lake Shore) Wednesday 07:00PM-09:30

Notes: This class satisfies the Engaged Learning requirement in the Undergraduate Research category. Combined with COMP 412-001.

COMP 313: Intermediate Object-Oriented Development

Section 001 (3687) Credits: 3; In person; Lecture
Instructor: Robert Yacobellis

Dumbach Hall:231 (Lake Shore) Tuesday, Thursday 01:00PM-02:15

COMP 317: Social, Legal, and Ethical Issues in Computing

Section 001 (5608) Credits: 3; Online; Lecture

Instructor: Matthew Paul Butcher

Online Times: TBA

Notes: This is an online class. All lectures will be pre-recorded. Students are asked to attend smaller-group online interactive discussions at regular intervals during the semester, with possible times chosen to fit different groups' schedules.

COMP 317: Social, Legal, and Ethical Issues in Computing

Section 01W (4465) Credits: 3; In person; Lecture

Instructor: Roxanne Schwab

Dumbach Hall:005 (Lake Shore) Wednesday 06:00PM-08:30

Notes: This is a writing intensive class. This class is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 317: Social, Legal, and Ethical Issues in Computing [Term: Eight Week - Second]

Section 700N (6010) Credits: 3; Online; Lecture

Instructor: Bruce A Montes

Online Tuesday 07:30PM-09:00

Notes: COMP 317-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 317-700N is an online class. Required synchronous sessions will be held 7:30-9PM CST

COMP 320: Software Systems Analysis [Term: Eight Week - First]

Section 700N (5979) Credits: 3; Online; Lecture

Instructor: Sargon Hasso

Online Wednesday 06:00PM-09:00

Notes: COMP 320-700N is reserved for students in the School of Continuing & Professional Studies. For more information go to http://www.luc.edu/adult-education/admission/nonscps-enrollment/ COMP 320-700N is an online section. Required synchronous sessions will be held Wednesday, 6-9PM CST.

COMP 324: Client-Side Web Development

Section 001 (5609) Credits: 3; In person; Lecture

Instructor: Nicholas J Hayward

Corboy Law Center:0301 (Water Tower) Monday 04:15PM-06:45

Notes: Combined with COMP 424-001

COMP 339: Distributed Systems

Section 001 (5610) Credits: 3; In person; Lecture

Instructor: Sarah Kaylor

Dumbach Hall:229 (Lake Shore) Wednesday 07:00PM-09:30

Notes: Combined with COMP 439-001

COMP 340: Computer Forensics

Section 001 (5611) Credits: 3; Online; Lecture

Instructor: Thomas Yarrish

Online Times: TBA

Notes: This is an online class. Synchronous meeting time: Tuesday, 7:00 pm - 9:30 pm. Combined with COMP 488-340.

COMP 341: Human-Computer Interaction

Section 001 (5612) Credits: 3; In person; Lecture

Instructor: Nicholas J Hayward

Corboy Law Center:0711 (Water Tower) Thursday 07:00PM-09:30

Notes: Combined with COMP 441-001

COMP 348: Network Security

Section 001 (5613) Credits: 3; In person; Lecture

Instructor: Corby Schmitz

Corboy Law Center:L08 (Water Tower) Friday 05:30PM-08:00

Notes: Combined with COMP 448-001

COMP 348: Network Security

Section 002 (5614) Credits: 3; Online; Lecture

Instructor: Corby Schmitz

Online Times: TBA

Notes: This is an online class. The classroom session will be broadcast live on Friday evenings via Adobe-Connect, allowing online student interaction. Sessions will also be recorded and made available. Students may participate synchronously or asynchronously at their discretion. Combined with COMP 448-002.

COMP 353: Database Programming

Section 001 (3690) Credits: 3; In person; Lecture

Instructor: Channah Naiman

Cuneo Hall:116 (Lake Shore) Monday 04:15PM-06:45

COMP 363: Design and Analysis of Computer Algorithms

Section 001 (4466) Credits: 3; In person; Lecture

Instructor: Mark Albert

Mundelein Center:0507 (Lake Shore) Thursday 02:45PM-05:15

Notes: This course is restricted to undergraduate students. Graduate students wishing to enroll in a section of this course should contact their departmental graduate advisor.

COMP 372: Programming Languages

Section 001 (5616) Credits: 3; In person; Lecture

Instructor: Konstantin Laufer

Mundelein Center:0607 (Lake Shore) Tuesday 02:45PM-05:15

Notes: Combined with COMP 471-001

COMP 373: Objects, Frameworks, and Patterns

Section 001 (5617) Credits: 3; In person; Lecture

Instructor: Berhane Zewdie

Corboy Law Center:0711 (Water Tower) Tuesday 07:00PM-09:30

Notes: Combined with COMP 473-001

COMP 378: Artificial Intelligence

Section 001 (5618) Credits: 3; In person; Lecture

Instructor: Conrad Weisert

Corboy Law Center:0303 (Water Tower) Monday 02:45PM-05:15

Notes: Combined with COMP 484-001

COMP 383: Computational Biology

Section 001 (5619) Credits: 4; In person; Lecture

Instructor: Heather E. Wheeler

Dumbach Hall:228 (Lake Shore) Monday 02:45PM-04:05

Cuneo Hall:202 (Lake Shore) Wednesday 02:45PM-04:45

Notes: Combined Section ID:

Combined with COMP 488-383

COMP 384: Metagenomics

Section 01E (6266) Credits: 3; In person; Lecture

Instructor: Michael Bradley Burns

Life Science Buildin:212 (Lake Shore) Monday, Wednesday, Friday 09:20AM-10:10

Notes: Combined Section ID:

COMP 384 is permanently cross-listed with BIOL 392. Register under BIOL 392. This class satisfies the Engaged Learning requirement in the Undergraduate Research category.

COMP 388: Topics in Computer Science: **Physical Design & Fabrication**

Section 007 (5620) Credits: 3; In person; Lecture

Instructor: Jonathan Durston

Crown Center:105 (Lake Shore) Monday 07:00PM-09:30

Notes: Combined with COMP 488-007

COMP 388 Topic: Game Design and Development

Section 323 (6375) Credits: 3; In person; Lecture

Instructor: Nicholas J Hayward

Cuneo Hall:002 (Lake Shore) Wednesday 02:45PM-05:15

Description similar to: comp323

COMP 388 Topic: Robotics Software Development

Section 367 (6027) Credits: 3; In person; Lecture

Instructor: William Honig

Mundelein Center:0608 (Lake Shore) Thursday 01:30PM-04:00

Old Syllabus: http://people.cs.luc.edu/whonig/comp-388-488-robotics/Comp388488ROBOTICSyllabus.pdf.

Notes: Combined with COMP 488-367

COMP 391: Internship in Computer Science

Section 01E (2162) Credits: 1 - 6; In person; Field Studies

Instructor: Ronald I Greenberg, Robert Yacobellis

Place TBA (Lake Shore) Times: TBA

Notes: This class satisfies the Engaged Learning requirement in the Internship category. Department Consent Required.

COMP 391: Internship in Computer Science

Section 11E (6049) Credits: 1 - 6; Online; Field Studies

Instructor: Ronald I Greenberg, Robert Yacobellis

Online Times: TBA

Notes: This class satisfies the Engaged Learning requirement in the Internship category. Department Consent Required.

COMP 397: Research Seminar

Section 001 (4284) Credits: 1; In person; Seminar

Instructor: Mark Albert

Mundelein Center:0507 (Lake Shore) Tuesday 04:00PM-05:15

COMP 397: Research Seminar **[Term: Seven Week -Third]**

Section 002 (6068) Credits: 1; In person; Seminar

Instructor: Mark Albert

OTHER: (Lake Shore) Tuesday 04:00PM-05:15

Notes: This class meets in MUND 507 with COMP 397-001.

COMP 398: Independent Study **1-6 credits** You cannot register yourself for an independent study course! You must find a faculty member who agrees to supervisor the work that you outline and schedule together. This *supervisor arranges to get you registered*. Possible supervisors are: Dmitriy Dligach, Peter L Dordal, Ronald I Greenberg, Nicholas J Hayward, William Honig, Konstantin Laufer, Channah Naiman, Maria Del Carmen Saenz, Chandra N Sekharan, George Thiruvathukal, Heather E. Wheeler, Robert Yacobellis

1.3.2 Graduate Courses

COMP 412: Free/Open Source Computing

Section 001 (5621) Credits: 3; In person; Lecture

Instructor: Michael Lewis

Crown Center:105 (Lake Shore) Wednesday 07:00PM-09:30

Notes: Combined with COMP 312-01E

COMP 413: Intermediate Object-Oriented Development

Section 001 (3692) Credits: 3; In person; Lecture

Instructor: Robert Yacobellis

School of Communicat:013 (Water Tower) Tuesday 04:15PM-06:45

COMP 417: Social, Legal, and Ethical Issues in Computing

Section 001 (3693) Credits: 3; In person; Lecture

Instructor: Peter L Dordal

School of Communicat:010 (Water Tower) Thursday 04:15PM-06:45

COMP 418: Combinatorial Mathematics

Section 001 (5469) Credits: 3; In person; Lecture

Instructor: William C Huffman

Dumbach Hall:006 (Lake Shore) Tuesday, Thursday 10:00AM-11:15

Notes: COMP 418 is cross listed with MATH 418. Please register for MATH 418

COMP 424: Client-Side Web Design

Section 001 (5623) Credits: 3; In person; Lecture

Instructor: Nicholas J Hayward

Corboy Law Center:0301 (Water Tower) Monday 04:15PM-06:45

Notes: Combined with COMP 324-001

COMP 439: Distributed Systems

Section 001 (5624) Credits: 3; In person; Lecture

Instructor: Sarah Kaylor

Dumbach Hall:229 (Lake Shore) Wednesday 07:00PM-09:30

Notes: Combined with COMP 339-001

COMP 441: Human-Computer Interface Design

Section 001 (5625) Credits: 3; In person; Lecture

Instructor: Nicholas J Hayward

Corboy Law Center:0711 (Water Tower) Thursday 07:00PM-09:30

Notes: Combined with COMP 341-001

COMP 448: Network Security

Section 001 (5626) Credits: 3; In person; Lecture

Instructor: Corby Schmitz

Corboy Law Center:L08 (Water Tower) Friday 05:30PM-08:00

Notes: Combined with COMP 348-001

COMP 448: Network Security

Section 002 (5627) Credits: 3; Online; Lecture

Instructor: Corby Schmitz

Online Times: TBA

Notes: This is an online class. The classroom session will be broadcast live on Friday evenings via Adobe-Connect, allowing online student interaction. Sessions will also be recorded and made available. Students may participate synchronously or asynchronously at their discretion. Combined with COMP 348-002.

COMP 460: Algorithms and Complexity

Section 001 (4297) Credits: 3; In person; Lecture

Instructor: Ronald I Greenberg

Cuneo Hall:218 (Lake Shore) Wednesday 02:45PM-05:15

COMP 471: Programming Languages

Section 001 (5630) Credits: 3; In person; Lecture

Instructor: Konstantin Laufer

Mundelein Center:0607 (Lake Shore) Tuesday 02:45PM-05:15

Notes: Combined with COMP 372-001

COMP 473: Object-Oriented Programming

Section 001 (5631) Credits: 3; In person; Lecture

Instructor: Berhane Zewdie

Corboy Law Center:0711 (Water Tower) Tuesday 07:00PM-09:30

Notes: Combined with COMP 373-001

COMP 474: Software Engineering

Section 001 (3694) Credits: 3; Online; Lecture
Instructor: Christopher Stone
Online Times: TBA

Notes: This is an online class. Synchronous meeting time: Wednesday, 7:00 pm - 9:30 pm.

COMP 484: Artificial Intelligence

Section 001 (5632) Credits: 3; In person; Lecture
Instructor: Conrad Weisert
Corboy Law Center:0303 (Water Tower) Monday 02:45PM-05:15

Notes: Combined with COMP 378-001

COMP 488: Topics in Computer Science: **Physical Design & Fabrication**

Section 007 (5633) Credits: 3; In person; Lecture
Instructor: Jonathan Durston
Crown Center:105 (Lake Shore) Monday 07:00PM-09:30

Notes: Combined with COMP 388-007

COMP 488 Topic: Data Warehouse & Mining

Section 301 (5634) Credits: 3; In person; Lecture
Instructor: Channah Naiman
Corboy Law Center:0201 (Water Tower) Wednesday 04:15PM-06:45
Description similar to: *COMP 300: Data Warehousing and Data Mining*

Notes: Combined with COMP 300-001

COMP 488 Topic: Data Warehouse & Mining

Section 302 (5635) Credits: 3; Online; Lecture
Instructor: Channah Naiman
Online Times: TBA
Description similar to: *COMP 300: Data Warehousing and Data Mining*

Notes: This class is a totally online, asynchronous course. Exams may be synchronous and in person. Combined with COMP 300-002.

COMP 488 Topic: Database Administration

Section 305 (5636) Credits: 3; In person; Lecture
Instructor: Peter L Dordal
School of Communicat:010 (Water Tower) Tuesday 04:15PM-06:45
Description similar to: *COMP 305: Database Administration*

Notes: Combined with COMP 305-001

COMP 488 Topic: Game Design and Development

Section 323 (6376) Credits: 3; In person; Lecture
Instructor: Nicholas J Hayward
Cuneo Hall:002 (Lake Shore) Wednesday 02:45PM-05:15
Description similar to: comp323

COMP 488 Topic: Comp Forensics Investigations

Section 340 (5637) Credits: 3; Online; Lecture
Instructor: Thomas Yarrish

Online Times: TBA

Description similar to: *COMP 340: Computer Forensics*

Notes: This is an online class. Synchronous meeting time: Tuesday, 7:00 pm - 9:30 pm. Combined with COMP 340-001.

COMP 488 Topic: Robotics Software Development

Section 367 (6028) Credits: 3; In person; Lecture
Instructor: William Honig
Mundelein Center:0608 (Lake Shore) Thursday 01:30PM-04:00
Old Syllabus: http://people.cs.luc.edu/whonig/comp-388-488-robotics/Comp388488ROBOTICSyllabus.pdf.

Notes: Combined with COMP 388-367

COMP 488 Topic: Computational Biology

Section 383 (5639) Credits: 3; In person; Lecture
Instructor: Heather E. Wheeler
Dumbach Hall:228 (Lake Shore) Monday 02:45PM-04:05
Cuneo Hall:202 (Lake Shore) Wednesday 02:45PM-04:45
Description similar to: *COMP 383: Computational Biology*

Notes: Combined with COMP 383-001

COMP 488: Topics in Computer Science: **Organizational Change and Development**

Section 472 (6029) Credits: 3; In person; Lecture
Instructor: Guy Bevente
Corboy Law Center:0423 (Water Tower) Monday 07:00PM-09:30

Notes: Organizational Change and Development

COMP 490: Independent Project **1-6 credits** You cannot register yourself for an independent study course! You must find a faculty member who agrees to supervisor the work that you outline and schedule together. This *supervisor arranges to get you registered.* Possible supervisors are: Konstantin Laufer, George Thiruvathukal

COMP 499: Internship

Section 001 (2177) Credits: 1 - 6; In person; Independent Study
Instructor: Konstantin Laufer, Chandra N Sekharan
Place TBA (Water Tower) Times: TBA

Notes: This course involves an internship experience. Department Consent Required.

COMP 605: Master of Science Study

Section 001 (2694) Credits: 0; In person; FTC-Supervision
Instructor: Andrew N Harrington, Konstantin Laufer
Place TBA (Water Tower) Times: TBA

Notes: Department Consent Required.

If you wish to look forward further, see

- longtermschedule
- cuneo2yrschedule

Or look further back:

- schedulearchive

Tutoring

- http://luc.edu/cs/schedules/tutoringhours/

UNDERGRADUATE DEGREE PROGRAMS

2.1 B.S. in Bioinformatics

2.1.1 Overview

The field of Bioinformatics is rooted in the massive databases generated by worldwide DNA sequencing projects and by ever widening 3-dimensional protein structure studies. Mining the wealth of biological and chemical information contained in these databases requires the tools of computer science and statistics. Bioinformatics approaches have already led to countless scientific, medical, and agricultural breakthroughs that would have taken decades to achieve without the foresight of a small number of visionary scientists and programmers. This major provides students with the training, skills, and opportunity to become charter members of this exciting, ground-breaking discipline with virtually limitless post-graduate educational and career advancement possibilities for years to come. Loyola is home to a superb group of faculty members already working at the interfaces of these four disciplines, who are combining their expertise to offer students this unique educational opportunity. This interdisciplinary major has its own detailed web pages at http://www.luc.edu/bioinformatics.

2.1.2 Curriculum

Please see http://www.luc.edu/bioinformatics/academics_bs.shtml.

2.2 B.S. in Communication Networks and Security

2.2.1 Overview

This new major grows out of the enormous importance of network computing and the major challenges to security that these networks pose. Students examine the architecture, properties, management and performance of both wired and wireless networks, including how to keep them reliable and secure. Students gain the talents and skills necessary for success in today's organizations according to current industry practices: planning, designing, implementing and administering voice and data communication networks; assessing and implementing the communication and security requirements of an organization in the form of a secure communication infrastructure; and functioning as an effective member of a network and security services division in an organization.

The Department of Labor projects a 57% (106,000) increase in the demand for Network systems and data communications analysts and a 37% increase (94,000) in Network and computer systems administrators over a 10 years period.

2.2.2 Curriculum

If you enrolled prior to Fall 2013 you may use the old Communication Networks and Security B.S. curriculum located here.

Major Requirements (48 Credit Hours)

One of the following must be taken:

- MATH 131: Applied Calculus I
- MATH 161: Calculus I

One of the following must be taken:

- *COMP 170: Introduction to Object-Oriented Programming*
- *COMP 215: Object-Oriented Programming with Mathematics*

One of the following must be taken:

- *COMP 264: Introduction to Computer Systems*
- *COMP 271: Data Structures*

All of the following must be taken:

- *COMP 163: Discrete Structures*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- *COMP 343: Introduction to Computer Networks*
- *COMP 347: Intrusion Detection and Computer Security*
- *COMP 348: Network Security*

Practicum Capstone

Six (6) credits taken from one or more of *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science*, and *COMP 398: Independent Study*. See the details of registering in the links for each course. (See also individual degree requirements, which generally permit three additional units beyond the practicum to be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.) Students are encouraged to complete these credits during junior and senior years to draw on prior experience.

Electives

18 credits taken from:

- *COMP 250: Introduction to Scientific and Technical Communication* or ENGL 210: Business Writing
- *COMP 264: Introduction to Computer Systems* or *COMP 271: Data Structures* NOTE: You must take one of these classes as part of the Major requirements. The second one can be used as an elective if taken.
- isscm349
- Any COMP 300 level courses NOTE: A special case is *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science* and *COMP 398: Independent Study* : Three additional units beyond the practicum can be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.

Suggested Ordering of Courses

Year 1

- MATH 131: Applied Calculus I or MATH 161: Calculus I
- *COMP 163: Discrete Structures*
- *COMP 170: Introduction to Object-Oriented Programming*

Year 2

- *COMP 264: Introduction to Computer Systems* or *COMP 271: Data Structures*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- Electives

Year 3

- *COMP 343: Introduction to Computer Networks*
- *COMP 347: Intrusion Detection and Computer Security*
- *COMP 348: Network Security*
- Electives

Year 4

- Electives
- Practicum

2.3 B.S. in Computer Science

2.3.1 Overview

The B.S. degree program in computer science provides a balance between theoretical foundations and applied computer science with the ultimate goal of presenting knowledge likely to be of ongoing value throughout one's career. The preparation of the B.S. is aimed at students who want pursue a career as an industry practitioners and/or as an academic.

2.3.2 Curriculum

If you enrolled prior to Fall 2013 you may use the old Computer Science B.S. curriculum located here.

Major Requirements (51 Credit Hours)

Both of the following must be taken:

- MATH 161: Calculus I
- MATH 162: Calculus II

One of the following must be taken:

- *COMP 170: Introduction to Object-Oriented Programming*
- *COMP 215: Object-Oriented Programming with Mathematics*

One of the following must be taken:

- *COMP 372: Programming Languages*
- *COMP 374: Introduction to Operating Systems*

All of the following must be taken:

- *COMP 163: Discrete Structures*
- *COMP 264: Introduction to Computer Systems*
- *COMP 271: Data Structures*
- *COMP 313: Intermediate Object-Oriented Development*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- *COMP 363: Design and Analysis of Computer Algorithms*

Practicum Capstone

Six (6) credits taken from one or more of *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science*, and *COMP 398: Independent Study*. See the details of registering in the links for each course. (See also individual degree requirements, which generally permit three additional units beyond the practicum to be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.) Students are encouraged to complete these credits during junior and senior years to draw on prior experience.

Electives

13 Credits taken from:

- *COMP 250: Introduction to Scientific and Technical Communication* or ENGL 210: Business Writing
- isscm349
- Any COMP 300 level electives NOTE: A special case is *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science* and *COMP 398: Independent Study* : Three additional units beyond the practicum can be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.

Suggested Ordering of Courses

Year 1

- MATH 161: Calculus I and MATH 162: Calculus II
- *COMP 163: Discrete Structures*
- *COMP 170: Introduction to Object-Oriented Programming*

Year 2

- *COMP 264: Introduction to Computer Systems*
- *COMP 271: Data Structures*
- *COMP 313: Intermediate Object-Oriented Development*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*

Year 3

- *COMP 372: Programming Languages*
- *COMP 363: Design and Analysis of Computer Algorithms*
- Electives

Year 4

- *COMP 374: Introduction to Operating Systems*
- Electives
- Practicum

2.4 B.S. in Mathematics and Computer Science

2.4.1 Overview

Many parts of computer science, including scientific computing, analysis of algorithms, and advanced research, use much mathematics. This major is for those people who want this highly analytical direction.

2.4.2 Curriculum

Math Requirements

Two years of Calculus

- MATH 161: Calculus I
- MATH 162: Calculus II
- MATH 263: Multivariable Calculus
- MATH 264: Ordinary Differential Equations

Including Multivariable Calculus, and Differential Equations.

- MATH 201: Elementary Number Theory
- MATH 212: Linear Algebra
- MATH 313: Abstract Algebra
- MATH 351: Introduction to Real Analysis I
- MATH 304: Probability and Statistics I / STAT 304: Probability and Statistics I

Two of the following five courses:

- MATH 309: Numerical Methods / *COMP 309: Numerical Methods*
- MATH 314: Advanced Topics in Abstract Algebra
- MATH 315: Advanced Topics in Linear Algebra
- MATH 352: Introduction to Real Analysis II
- MATH 353: Introductory Complex Analysis

Computer Science Requirements

- *COMP 150: Introduction to Computing* (may be replaced by a 300-level classroom elective if *COMP 215: Object-Oriented Programming with Mathematics* is taken)
- Introduction to Object-Oriented Programming & Data Structures
  - Either *COMP 170: Introduction to Object-Oriented Programming*
  - OR *COMP 215: Object-Oriented Programming with Mathematics*
- *COMP 264: Introduction to Computer Systems*
- *COMP 271: Data Structures*

One of the following three courses:

- *COMP 313: Intermediate Object-Oriented Development*
- *COMP 363: Design and Analysis of Computer Algorithms*
- *COMP 376: Formal Languages and Automata*

One of the following six courses:

- *COMP 336: Markup Languages*
- *COMP 337: Introduction to Concurrency*
- *COMP 338: Server-Based Software Development*
- *COMP 339: Distributed Systems*
- *COMP 353: Database Programming*
- *COMP 373: Objects, Frameworks, and Patterns*

Electives

- Any two 300-level, 3-credit courses in Computer Science.

2.5 B.S. in Physics and Computer Science

2.5.1 Overview

Physics is understood in terms of many mathematical relationships that are much easier to state than solve, and computer science has become a major part of many physicists' work to solve enormous problems. This major is preparation for graduate study in physics, applied physics, computer science, and especially in the burgeoning research field of computational physics, as well as in many branches of engineering. Employment opportunities are in industry, R&D and manufacturing, research and teaching in academic institutions, and research in government and private laboratories.

2.5.2 Curriculum

Math Requirements

Two years of Calculus

- MATH 161: Calculus I

- MATH 162: Calculus II
- MATH 263: Multivariable Calculus
- MATH 264: Ordinary Differential Equations

Computer Science Requirements

- *COMP 150: Introduction to Computing* (may be replaced by a 300-level classroom elective if *COMP 215: Object-Oriented Programming with Mathematics* is taken)
- Introduction to Object-Oriented Programming & Data Structures
    - Either *COMP 170: Introduction to Object-Oriented Programming*
    - OR *COMP 215: Object-Oriented Programming with Mathematics*
- *COMP 264: Introduction to Computer Systems*
- *COMP 271: Data Structures*
- *COMP 313: Intermediate Object-Oriented Development* OR *COMP 363: Design and Analysis of Computer Algorithms* OR *COMP 376: Formal Languages and Automata*

One course from the following list of programming-intensive CS courses:

- *COMP 336: Markup Languages*
- *COMP 337: Introduction to Concurrency*
- *COMP 338: Server-Based Software Development*
- *COMP 339: Distributed Systems*
- *COMP 353: Database Programming*
- *COMP 373: Objects, Frameworks, and Patterns*

Electives

- Any three 300-level, 3-credit courses in Computer Science.

Physics Requirements

- PHYS 125: General Physics I
- PHYS 126: General Physics II
- PHYS 135: General Physics Lab I
- PHYS 136: General Physics Lab II
- PHYS 235: Modern Physics
- PHYS 237: Modern Physics Lab
- PHYS 303: Electronics I
- PHYS 310: Optics
- PHYS 314: Theoretical Mechanics I
- PHYS 328: Thermal Physics & Statistical Mechanics

• PHYS 351: Electricity & Magnetism I

2.6 B.S. in Information Technology

2.6.1 Overview

This major prepares students who plan to design, create, and administer large information bases used by organizations.

Enterprises have an ever-growing investment in the exploding quantity of information, especially in web related data, that requires increasingly sophisticated approaches for efficient access and productive use. Students gain the talents and skills to be successful in today's organizations following current industry practices: planning, designing, implementing and administering data information and knowledge bases that can be effectively mined; assessing the information and data requirements of an organization and implementing these requirements as an information system; and functioning as an effective member of an information services division in an organization.

2.6.2 Curriculum

If you enrolled prior to Fall 2013 you may use the old Information Technology B.S. Curriculum located here.

Major Requirements (48 Credit Hours)

One of the following must be taken:

• STAT 103: Fundamentals of Statistics
• STAT 203: Statistics
• ISSCM 241: Business Statistics
• PSYC 304: Statistics

One of the following must be taken:

• *COMP 125: Visual Information Processing*
• *COMP 150: Introduction to Computing*
• *COMP 170: Introduction to Object-Oriented Programming*
• *COMP 215: Object-Oriented Programming with Mathematics*

One of the following must be taken:

• *COMP 251: Introduction to Database Systems*
• *COMP 271: Data Structures*

All of the following must be taken:

• *COMP 163: Discrete Structures*
• *COMP 300: Data Warehousing and Data Mining*
• *COMP 305: Database Administration*
• *COMP 317: Social, Legal, and Ethical Issues in Computing*
• *COMP 353: Database Programming*
• *COMP 377: IT Project Management* or isscm349

Practicum Capstone

Six (6) credits taken from one or more of *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science*, and *COMP 398: Independent Study*. See the details of registering in the links for each course. (See also individual degree requirements, which generally permit three additional units beyond the practicum to be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.) Students are encouraged to complete these credits during junior and senior years to draw on prior experience.

Electives

15 credits taken from:

- *COMP 250: Introduction to Scientific and Technical Communication* or ENGL 210: Business Writing
- *COMP 251: Introduction to Database Systems* or *COMP 271: Data Structures* NOTE: You must take one of these classes as part of the Major requirements. The second one can be used as an elective if taken.
- *COMP 264: Introduction to Computer Systems*
- MGMT 304: Strategic Management
- MGMT 315: International Management
- MGMT 318: Organizational Development and Change
- MGMT 320: Leading and Managing Team
- MGMT 335: Micro-enterprise Consulting
- MGMT 360: Values Based Leadership
- ENTR 310: Innovation and Entrepreneurship
- Any COMP 300 level electives NOTE: A special case is *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science* and *COMP 398: Independent Study* : Three additional units beyond the practicum can be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.
- MGMT 201 or ACCT 201 may count as 3 credits of major Electives if either ISSCM 349 or ENTR 310 is completed to count toward this major.

Suggested Ordering of Courses

Year 1

- ISSCM 241: Business Statistics or (PSYC 304: Statistics after core PSYC 101: General Psychology)
- *COMP 163: Discrete Structures*
- *COMP 125: Visual Information Processing* or *COMP 150: Introduction to Computing* or *COMP 170: Introduction to Object-Oriented Programming*

Year 2

- *COMP 251: Introduction to Database Systems* or *COMP 271: Data Structures*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- Electives

Year 3

- *COMP 300: Data Warehousing and Data Mining*
- *COMP 305: Database Administration*
- *COMP 353: Database Programming*
- Electives

Year 4

- Electives
- Practicum

2.7 B.S. in Software Engineering

2.7.1 Overview

With software applications of enormous size, complexity, and expense now prevalent in diverse domains, software engineering has never been as important a field as it is now. Students gain necessary talents to be successful in today's organizations, following current industry practices: designing and developing software; understanding and applying software development processes and methodologies in their work; leveraging software development tools used in the various phases of the development life cycle; and functioning as an effective member of a software development team or organization. Students develop their knowledge and skill though high-level electives where they write major projects in diverse areas such as client/server programming for the web, distributed programming for large clusters of processors, database programming, and markup language transformation. While working on modern applications with current software engineering practices such as Extreme Programming, students learn to analyze and apply good algorithms and other relevant tools.

The department of Labor estimates there will be a 39% (184,000) increase in demand for Computer Systems Analysts in the next ten years. And in the recent projections of job growth by the National Association of Colleges and Employers (NACE), new hires are on the upswing for software design and development graduates in particular, with an average starting salary of $57,729.

2.7.2 Curriculum

This Degree was previously known as Software Development.

If you were enrolled prior to Fall 2013 you may use the curriculum for the old B.S. Software Development Degree, which can be found here.

Major Requirements (48 Credit Hours)

One of the following must be taken:

- MATH 131: Applied Calculus I
- MATH 161: Calculus I (4 credits)

One of the following must be taken:

- *COMP 170: Introduction to Object-Oriented Programming*
- *COMP 215: Object-Oriented Programming with Mathematics*

One of the following must be taken:

- *COMP 333: Formal Methods in Software Engineering*

- *COMP 373: Objects, Frameworks, and Patterns*

All of the following must be taken:

- *COMP 163: Discrete Structures*
- *COMP 271: Data Structures*
- *COMP 313: Intermediate Object-Oriented Development*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- *COMP 330: Software Engineering*

Practicum Capstone

Six (6) credits taken from one or more of *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science*, and *COMP 398: Independent Study*. See the details of registering in the links for each course. (See also individual degree requirements, which generally permit three additional units beyond the practicum to be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.) Students are encouraged to complete these credits during junior and senior years to draw on prior experience.

Electives

18 credits taken from:

- *COMP 250: Introduction to Scientific and Technical Communication* or ENGL 210: Business Writing
- *COMP 264: Introduction to Computer Systems*
- isscm349
- Any COMP 300 level electives NOTE: A special case is *COMP 390: Broadening Participation in STEM (Computing, Math & Science)*, *COMP 391: Internship in Computer Science* and *COMP 398: Independent Study* : Three additional units beyond the practicum can be counted as an elective, as long as you take no more than 6 units of 391 and no more than 6 units of 398.

Suggested Ordering of Courses

Year 1

- MATH 131: Applied Calculus I or MATH 161: Calculus I
- *COMP 163: Discrete Structures*
- *COMP 170: Introduction to Object-Oriented Programming* or *COMP 215: Object-Oriented Programming with Mathematics*
- *COMP 271: Data Structures*

Year 2

- *COMP 313: Intermediate Object-Oriented Development*
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- Electives

Year 3

- *COMP 330: Software Engineering*

- *COMP 333: Formal Methods in Software Engineering* or *COMP 373: Objects, Frameworks, and Patterns*
- Electives

Year 4

- Electives
- Practicum

2.8 BS/MS Dual Degree Programs

2.8.1 Overview

The BS/MS programs in Information Technology, Software Engineering, and Computer Science are five-year programs that give academically successful Loyola undergraduates the opportunity to pursue the MS degree in Information Technology, Software Engineering, or Computer Science while completing their BS degree. The applicant can be pursuing any of the BS degrees offered by the Department of Computer Science, including the joint majors with Mathematics or Physics or the interdisciplinary Bioinformatics.

These programs reduce the total number of courses needed and the total time needed for the combined degrees.

Students may not automatically enroll in the BS/MS program during Freshman admission. They must specifically apply to the Graduate School, generally in their Junior year by using The Graduate Application Form. See *Application Requirements* and *Application Information* for more details.

2.8.2 Curriculum

The stand-alone MS programs each require 30 credits of graduate courses. This generally takes one and a half years to complete. BS/MS students must also take 30 credits of courses at the 400 level or greater. The main advantages come from the fact that some of the MS courses may be double counted, applying to the BS also, and there can be a shift in some of the required MS courses based on what students took in the BS portion. In particular students in the BS/MS program are allowed to "double count" 9 credits (generally three 3-credit courses) of their Loyola COMP courses for both their BS and MS degrees. Each must be a 400 level course taken in the student's senior year *in place* of a 300 level major course. In addition, a student wishing to take 123 or more credits while an undergraduate, can have 3 of the extra credits be for a fourth 400 level COMP course, and count that toward the MS if the extra course is need for *no* undergraduate requirement, from the university, college, core, major or minor. Hence a BS/MS student with 120 credits and 9 400-level COMP credits counted for the BS is required to take only 21 credits (instead of 30 credits) of additional 400-level courses. If a student takes a further 3-credit 400 level COMP course beyond the 120 credits needed for the BS, then only 18 further credits are needed. In either case students can finish the program in five years. Some further special rules:

1. All students need to take *COMP 271: Data Structures* as part of their undergraduate major.

2. Students pursuing an MS in Software Engineering or Computer Science must take *COMP 313: Intermediate Object-Oriented Development* by their senior year.

3. Students may take at most 6 credits total of internship at the undergraduate and graduate level, *COMP 391: Internship in Computer Science* plus *COMP 499: Internship*.

4. A student with credit for a 300 level COMP course that was taught in a combined class with a 400 level course may not take the 400 level course later for separate credit without permission from the Graduate Program Director. In particular for students with credit for *COMP 317: Social, Legal, and Ethical Issues in Computing*, the MS requirement for *COMP 417: Social, Legal, and Ethical Issues in Computing* is waived, to be replaced by 3 MS elective credits. Similarly, in programs requiring *COMP 413: Intermediate Object-Oriented Development*,

COMP 313: Intermediate Object-Oriented Development will satisfy the requirement, but still the course will need to be replaced by 3 MS elective credits.

5. In MS programs with restricted electives lists with two or more courses required from a larger explicit list, one of the courses may be waived if the corresponding 300 level undergraduate course was successfully completed at Loyola. The course must still be replaced by a 3-credit MS program elective.

6. All further individual MS program requirements must be met, but with the inclusion of the allowed 400 level courses taken as an undergraduate.

Application Requirements

The academic prerequisites to be considered for admission to the BS/MS programs in the Computer Science Department are as follows:

- Successful completion of a total of at least 15 credits of major COMP courses, including *COMP 271: Data Structures* (even if *COMP 271: Data Structures* is only an elective in the major) and including at least 9 credits of 300 level COMP courses. For students pursuing the BS degree in Computer Science and Mathematics, Computer Science and Physics, or Bioinformatics, the 300-level courses may alternately include up to 6 credits of 300 level major courses that are not COMP.

- A GPA of 3.5 or higher in all the major courses.

- A cumulative GPA of 3.3 or higher for all course work at Loyola;

- Satisfactory progress towards completion of Loyola's core.

Further notes on BS completion: Before the deadlines do apply to graduate with your B.S in the semester you will actually finish! Otherwise you complicate the conversion to graduate status. If the date when you will start graduate status changes from your original application, notify the GPD ahead of time so data in Locus can be fixed.

Application Information

Current Loyola students who have met the above academic prerequisites are encouraged to apply before registration opens for your senior year, though the program is still of practical help to a student who applies before the final undergrad semester in which s/he is taking COMP courses. Students who have an interest in the program are encouraged to consult with the Graduate Program Director for Computer Science, Dr. Andrew Harrington, in the semester prior to their application. Students who have substantial transfer or AP credit, or wish to finish the combined program in less than five years, should discuss the timing of their application with Dr. Harrington. The student should then fill out the on-line Graduate Application Form through the Graduate School web-site. The application will be evaluated upon completion of the following:

- The Graduate Application Form; applicants are to select the BS/MS option. Be careful of the misleading entry named "starting semester": The *starting semester for the program should be the first semester when the student will be in pure* **graduate** *status* (after expected completion of the BS). This is *not* generally the semester coming right after the student has filled out the application. This is *after* all double-counted courses are completed.

- A one page personal statement of purpose;

- Three letters of recommendation, at least two coming from faculty in the Department of Computer Science at Loyola University.

Note that the GRE exam is not required for applicants to the BS/MS program, even if they are applying for an assistantship. There is no application fee for applicants who apply on-line. At the time of consideration for admission, the student's most recent Loyola transcript will be examined. If the applicant has transfer credits from another college or university, the student may be required to submit those as well.

Requirements for Completion of the BS/MS Program

After admission to the BS/MS program, a student may request to be registered in their senior year for up to four 400 level COMP courses, with three of them replacing undergraduate major courses, and a fourth *if* student will graduate with at least 123 credits and not need the fourth course as a part of *any* undergraduate requirement. Students generally take the remainder of their 30 credits of 400-level classes during their fifth year. This schedule can be modified for students with AP or transfer credit who desire to finish the combined program in less than five years. Students in the program are expected to consult regularly with the Graduate Program Director to insure that they are on track for completion of both the BS and MS degrees. The following are required to complete a BS/MS degree program:

- Successful completion of one of the BS degrees in the Department of Computer Science or Bioinformatics or a joint major with Mathematics or Physics.

- The 30 credits of 400-level graduate courses, including those counted while an undergraduate, completed with a GPA of 3.0 or higher

Further Information

If you have additional questions about the program, please contact Dr. Andrew Harrington, Graduate Program Director for Computer Science. He can be reached at gpd@cs.luc.edu.

2.9 Minor in Information Technology

2.9.1 Overview

This minor is designed for students seeking understanding of important aspects of computing such as databases, networking, and/or security, especially from a managerial viewpoint. This may be a particularly valuable enhancement to a major in the School of Business as well as to majors in other fields in the College of Arts and Sciences or other divisions of the University..

2.9.2 Curriculum

Eighteen (18) total credits (or 21 if taking ACCT201 and MGMT201):

- *COMP 125: Visual Information Processing* or *COMP 150: Introduction to Computing* or *COMP 170: Introduction to Object-Oriented Programming* or *COMP 215: Object-Oriented Programming with Mathematics*.

- *COMP 251: Introduction to Database Systems* or *COMP 264: Introduction to Computer Systems* or *COMP 271: Data Structures*.

- *COMP 377: IT Project Management* or isscm349.

- One of *COMP 300: Data Warehousing and Data Mining*, *COMP 305: Database Administration*, *COMP 353: Database Programming*, *COMP 343: Introduction to Computer Networks*, *COMP 346: Introduction to Telecommunications*, *COMP 348: Network Security*, *COMP 349: Wireless Networks and Security*, *COMP 351: Network Management*, *COMP 340: Computer Forensics*, *COMP 347: Intrusion Detection and Computer Security*.

- Two more courses from the 200-level and 300-level courses listed above. One of these courses can be replaced by ACCT201 and MGMT201.

2.10 Minor in Computer Science

2.10.1 Overview

Computers are extremely pervasive in the modern world, and important connections have been established between computer science and virtually every other field of study. Thus, a computer science minor is a valuable enhancement to majors in other fields, whether in the sciences, social sciences, arts, humanities, business, etc.

(More specifically focused minors are also offered in *Minor in Information Technology* and *Minor in Computer Crime and Forensics*. In addition, majors are offered in the interdisciplinary fields of Bioinformatics, *B.S. in Mathematics and Computer Science*, and *B.S. in Physics and Computer Science*, as well as in *B.S. in Computer Science*, *B.S. in Communication Networks and Security*, *B.S. in Information Technology*, and *B.S. in Software Engineering*.)

2.10.2 Curriculum

- *COMP 170: Introduction to Object-Oriented Programming*
- *COMP 271: Data Structures*
- *COMP 125: Visual Information Processing* or *COMP 150: Introduction to Computing* or *COMP 163: Discrete Structures* or 3 credits of a 300-level COMP course
- *COMP 251: Introduction to Database Systems* or *COMP 264: Introduction to Computer Systems* or 3 credits of a 300-level COMP course
- Six (6) units of 300-level Computer Science electives (for instance, two 3-credit courses)
    - Note: *COMP 391: Internship in Computer Science* is **not** allowed.

Curriculum (pre-Fall 2015)

If you declared the Computer Science Minor prior to Fall 2015, you may use the old curriculum below.

- *COMP 150: Introduction to Computing*
- *COMP 170: Introduction to Object-Oriented Programming*
- *COMP 271: Data Structures*
- Nine (9) units of Computer Science electives (for instance, three 3-credit courses)
    - Note: 6 units at the 300 level and 3 units at the 200 or 300 level. *COMP 391: Internship in Computer Science* is **not** allowed.

2.11 Minor in Computer Crime and Forensics

2.11.1 Overview

Crimes in general are dealt with using preventive and detective techniques. Computer crime is no different. With the dramatic increase in the use of computers, networks, and the Internet, crimes committed via computers and networks have also risen rather sharply. Computer crime prevention falls in the areas known as computer and network security. Computer crime detection is generally known as computer forensics. Computer forensic investigations deal with white collar crime, telecommunications fraud, network intrusion detection, and criminal procedures. A computer forensic expert can help solve cases in money laundering, intellectual property, child pornography, embezzlement, e-mail harassment, murder, and terrorism.

Computer Crime and Forensics is an interdisciplinary minor of the Computer Science and Criminal Justice Departments that will impart knowledge to students in the areas of criminal justice system, courts, law and procedures, computer software, hardware, networks, and investigative and evidence-gathering protocols.

The Computer Crime and Forensics minor does not require any programming background and will appeal to those who want to use computers to solve criminal or civil cases where the evidence is traceable via a computer network or storage. The minor should particularly appeal to computer science, criminal justice and forensic sciences majors. The minor is open for any Loyola student who has an analytical and investigative mind and who enjoys working with computer tools.

2.11.2 Career Opportunities

With the minor, a student can pursue the career path of a computer forensic examiner, or an electronic discovery specialist, or pursue a legal career specializing in criminal and civil law related to computers (IP, internal fraud, misappropriation of trade secrets etc.).A scan of jobs posted in Hotjobs.com and Monster.com reveals that computer forensics specialists are much in demand. Jobs are available across a wide variety of industry segments, pharmaceuticals, accounting firms, financial services, and law firms. Typical titles of jobs advertised in the online jobs websites are: technology litigation support specialist, incident management engineers, computer forensics manager, law enforcement officers, information security specialist, electronic discovery specialist etc.

2.11.3 Curriculum

All of the following must be taken:

- CJC 101: The Criminal Justice System
- *COMP 150: Introduction to Computing* or *COMP 170: Introduction to Object-Oriented Programming* or *COMP 215: Object-Oriented Programming with Mathematics* / MATH 215: Object Oriented Math Programming
- *COMP 340: Computer Forensics*
- *COMP 347: Intrusion Detection and Computer Security*

One course from the following:

- CJC 322: Criminal Courts and Law
- CJC 323: Criminal Procedure

One course from the following:

- *COMP 264: Introduction to Computer Systems* (requires *COMP 170: Introduction to Object-Oriented Programming* prerequisite)
- *COMP 317: Social, Legal, and Ethical Issues in Computing*
- *COMP 343: Introduction to Computer Networks* (requires *COMP 170: Introduction to Object-Oriented Programming* and *COMP 271: Data Structures* sequence as prerequisites)

UNDERGRADUATE COURSES

3.1 COMP 102: Web Design and Multimedia Publishing

This course introduces foundations of the world wide web technology, HTML, and multimedia publishing techniques. Topics include HTML syntax, CSS, XML, RSS, and various multimedia formats.

3.1.1 Credit Hours

3

3.1.2 Prerequisites

None

3.1.3 Description

An introduction to the basic components and tools used in developing pages for the World Wide Web. Topics will include HTML, HTML Standards and validation, Cascading Style Sheets (CSS), and scripting. Tools appropriate for each of these will be discussed. Other topics may include CGI programming and graphics editors. The course is taught in a laboratory environment with hands-on instruction.

3.1.4 Outcome

An understanding of the technologies behind web sites and the ability to use them effectively.

3.1.5 Syllabi

No recent syllabi available.

3.2 COMP 104: Computer Animation

The course introduces techniques for understanding and developing dynamic and interactive media by using sound, motion, images, and text. Relevant software knowledge areas are covered.

3.2.1 Credit hours

3

3.2.2 Prerequisites

None

3.2.3 Description

Computer animation of multimedia objects is increasingly used in web publishing and in projects for product marketing. The purpose of this course is to introduce interactive elements in multimedia creation using computer animation and using simple programming components for interactivity. The course will teach techniques to create dynamic and interactive content using Macromedia's Flash software. Examples of interactivity include the ability to change the animation upon user input, as in a flying plane unveiling a banner when the user touches it or upon the expiration of a timer, as in rain turning to snow in 5 seconds or other event-driven systems. The course is appropriate for anyone who is interested in using computers to design, and publish simple graphical animations including web developers, and graphic artists. The salient feature of the course is that very little programming knowledge is needed although students should be comfortable in using software tools and be able to think logically to write simple, programming segments in a special language called Actionscript.

3.2.4 Outcome

Ability to publish created animated media projects to the web in a process that involves understanding human interface design.

3.2.5 Syllabi

No recent syllabi available.

3.3 COMP 111: History of Computing

The social and organizational history of humanity is intricately entangled with the history of technology in general and the technology of information in particular. Advances in this area have often been closely involved in social and political transformations. While the contemporary period is often referred to by such names as the Computing and Information Age, this is the culmination of a series of historical transformations that have been centuries in the making. This course will provide a venue for students to learn about history through the evolution of number systems and arithmetic, calculating and computing machines, and advanced communication technology via the internet.

3.3.1 Credit Hours

3

3.3.2 Prerequisites

None

3.3.3 Description

This course will provide a venue for students to learn about history through the evolution of number systems and arithmetic, calculating and computing machines, and advanced communication technology via the Internet. Students who take this course will attain a degree of technological literacy while studying core historical concepts. Students who complete this course will learn the key vocabulary of the computing discipline, which is playing a significant role in modern human though and new media communications. The History of Computing will be organized around the historical perspective. The relationships between social organization, intellectual climate, and technology will be examined and stressed.

3.3.4 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Thiruvathukal	https://luc.box.com/s/f5wmc53zwpb7r9q1w1s46mpdq3a0ajmm

3.4 COMP 120: Introduction to Computer Applications

This course, intended for non-science majors, offers a hands-on introduction to the development, functions, and applications of computers. It includes weekly lab assignments.

3.4.1 Alias

Math 149

3.4.2 Credit Hours

3

3.4.3 Prerequisites

MATH 100: Intermediate Algebra or equivalent

3.4.4 Description

This course, intended for non-science majors, offers a hands-on introduction to the development, functions, and applications of computers. Topics include e-mail, exploring the Internet, spreadsheets, word processing, database, desktop software, statistical packages, and programming. Uses of computing technology in business and the arts. We also review ethical, security, and privacy issues as they relate to computers and the world today. There are also weekly lab assignments in this course.

3.4.5 Outcome

Experience with Internet tools, desktop publishing, spreadsheets, databases, statistical packages, and some programming, and with applications to business and the arts; an understanding of ethical, security, and privacy issues relating to computers and the Internet.

3.4.6 Syllabi

No recent syllabi available.

3.5 COMP 122: Introduction to Digital Music

Computers and digital tools have been seeping into the world of music, and this course aims to explore this newly formed territory.

3.5.1 Credit Hours

3

3.5.2 Prerequisites

Required – MATH 118: Precalculus (or equivalent)

Preferred – *COMP 150: Introduction to Computing*

3.5.3 Description

This course will be cross-listed with Music.

Computers and digital tools have been seeping into the world of music, and this course aims to explore this newly formed territory. This course is intended for both Music and Computer Science students who wish to learn more about electronic music, signal processing, and algorithmic music composition.

3.5.4 Outcome

- understanding of the physics of musical sound and digital audio
- hands-on applications of algorithmic music composition and musicology
- ability to design and render digital instruments
- student-chosen final digital composition or musicology project

3.5.5 Syllabi

No recent syllabi available.

3.6 COMP 125: Visual Information Processing

This course, intended primarily for non-majors, provides an introduction to computer programming using a language well-suited to beginning programmers and practical applications, e.g., Visual Basic.Net.

3.6.1 Credit Hours

3

3.6.2 Prerequisites

None

3.6.3 Description

An elementary introduction to programming using a language such as Visual Basic. Topics include variables, formatted input/output, arrays, looping, conditional execution, subroutines, functions, computer graphics, animation. Applications to other disciplines are stressed.

3.6.4 Outcome

Understanding of computer mechanisms for representing and analyzing numerical and logical information and the power of programmability; practical ability to implement useful computing tools.

3.6.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Channah Naiman	https://luc.box.com/s/9tiy2yyrxjv6b2n60bumsa3bvn9vqgia
Fall 2011	Dr. William Honig	https://luc.box.com/s/9tiy2yyrxjv6b2n60bumsa3bvn9vqgia
Fall 2011	Dr. Ron Greenberg	https://luc.box.com/s/9tiy2yyrxjv6b2n60bumsa3bvn9vqgia

3.7 COMP 150: Introduction to Computing

The world overflows with electronic data. This course introduces programming in a simple, powerful language like Python, with selection, repetition, functions, graphical effects, and dynamic interaction with the Internet, plus connections to lower level computer organization and computer implications in the wider world.

3.7.1 Credit Hours

3

3.7.2 Prerequisites

None

3.7.3 Description

This course provides a broad survey introducing the many layers of the computer science discipline, emphasizing the computer's role and limitations as a tool for describing, organizing, and manipulating information applicable to many disciplines. Topics include binary logic expressed in electronic circuitry, machine architecture, basic programming in the very accessible language Python, data organization, the potential and limitations of machines, and useful tools.

This course serves as a terminal course for students who want a one-course introduction to the field, as well as a preliminary course to upper-level computer science offerings.

3.7.4 Outcome

Empowerment to manage and transform masses of data; understanding of technical, societal, and ethical issues involved.

3.7.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Michael Lewis	https://luc.box.com/s/rqubwh2u96999jzzsoek4c28a990rouz
Spring 2016	Dr. Andrew Harrington	https://luc.box.com/s/0dm3l12rbwxw3mj8is314180na6e34uj
Fall 2015	Dr. Christopher P Stone	https://luc.box.com/s/tvghcr954e0dkmky7dk2go9r7opjat8i
Past Offerings		https://luc.box.com/s/atg3mq2isjogedvc6cxz343oi5ilzdjb

3.8 COMP 163: Discrete Structures

This course covers the mathematical foundations of computer science, including such topics as complexity of algorithms, modular arithmetic, induction and proof techniques, graph theory, combinatorics, Boolean algebra, logic circuits, and automata.

3.8.1 Credit Hours

3

3.8.2 Prerequisites

None

3.8.3 Description

Mathematical foundations of computer science. Topics include: complexity of algorithms; modular arithmetic; induction and proof techniques; graph theory; combinatorics; Boolean algebra and switching systems (including Karnaugh maps and the Quine-McCluskey method); symbolic logic and logic circuits; and an introduction to automata and formal languages.

3.8.4 Outcome

The student will be prepared for the study of advanced ideas in computer science, from cryptography to databases to algorithms to computer architecture.

3.8.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Victoria Noquez	https://luc.box.com/s/crpjpmflgefgfiej4gdamvz5rqc19hc5

3.9 COMP 170: Introduction to Object-Oriented Programming

This programming intensive course with its weekly lab component introduces basic concepts of object-oriented programming in a language such as Java.

3.9.1 Credit Hours

3

3.9.2 Prerequisites

Students are required to have taken MATH 117: College Algebra as a prerequisite or to have been placed in MATH 118: Precalculus or higher.

Alternatively, students can take any **one** of the following courses as a prerequisite or as a co-requisite:

- *COMP 163: Discrete Structures*
- *COMP 150: Introduction to Computing*
- MATH 118: Precalculus
- MATH 131: Applied Calculus I
- MATH 161: Calculus I

3.9.3 Description

This course is an introduction to the computer science major, covering basic concepts using the C# (C-Sharp) object-oriented(OO) programming language.

The course addresses the following questions:

- What is an algorithm?
- How does one write, debug, run ("execute"), and test an effective computer program?
- How does one convert an algorithm into a computer program?
- How does one judge a program?
- What does "object-oriented" mean?

Topics include: variables, data types,input/output, loops and repetition, choice, arrays, subprograms,classes/objects, OO principles, and recursion.

3.9.4 Course Goals

Upon successful completion of the course, the student will be able to:

1. Write good programs of small to medium size – programs that are correct, high quality, and use correct and appropriate Object Oriented Programming techniques.

2. Reuse classes and Application Programming Interfaces (APIs) developed by others, especially the standard library APIs.

3. Understand and recognize proper programming style and demonstrate making design decisions consistent with Object Oriented methodologies.

4. Be able to read, understand, and interpret programs written by others in the same language.

3.9.5 Outcome

Ability to take a problem, break it into parts, specify algorithms, and express a solution in terms of variables, data types, input/output, repetition, choice, arrays, subprograms, classes, and objects; ability to judge a good program.

3.9.6 Syllabi

Semester/Year	Instructor	URL
Spring 2015	Dr. Thiruvathukal	https://luc.box.com/s/muehwlt68me4x77lacrht2gfg8z345vs
Spring 2014	Dr. Yacobellis	https://luc.box.com/s/muehwlt68me4x77lacrht2gfg8z345vs
Spring 2013	Dr. Harrington	https://luc.box.com/s/muehwlt68me4x77lacrht2gfg8z345vs

3.10 COMP 171: Scripting Languages

Scripting languages are rapid prototyping languages that are used extensively. This course covers the principles, syntax and semantics of widely used scripting languages.

3.10.1 Credit Hours

1

3.10.2 Prerequisites

None

3.10.3 Description

This is a course suitable for anyone wishing to study programming computers without pain. After taking this course, students should (a) have an understanding of what scripting languages can do and when they are suitable for use, (b) be able to program in Perl (c) be able to program in Python, and (d) have an understanding of their applicability to scientific and business domains.

3.10.4 Outcome

Students will learn how a program can be put together quickly and efficiently to solve problems.

3.10.5 Syllabi

Semester/Year	Instructor	URL
Spring 2012	Dr. Honig	https://luc.box.com/s/c98xuesae7fj3u86ua9fbtk7khl1hn49

3.11 COMP 180: Computing and Data Analysis

3.11.1 Credit Hours

3

3.11.2 Description

Computing and Data Analysis for the Sciences. Scientific computing emphasizes data analysis and visualization in a scientific context - analyzing data quickly for understanding by the individual, sharing automated workflows with collaborators, and preparing results for later publication. This course will emphasize rapid, interactive, and reproducible collaborative analysis of data for scientific contribution. At the end of this course, students will be well versed in the use of a specific, interactive environment for data analysis (likely Python, R, or MATLAB as indicated in the course notes) for analyzing data and sharing results.

3.12 COMP 215: Object-Oriented Programming with Mathematics

This is an introductory programming course for students interested in mathematics and scientific computing. Students will program primarily in a general object-oriented language such as Python, with supplementary exercises in a computer algebra system. Examples will be drawn primarily from applications of calculus, elementary number theory, and cryptography.

3.12.1 Credit Hours

3

3.12.2 Prerequisites

MATH 132: Applied Calculus II or MATH 162: Calculus II or permission of Instructor

3.12.3 Description

This is an introductory programming course for students interested in mathematics and scientific computing. Students will program primarily in a general object-oriented language such as Python, with supplementary exercises in a computer algebra system. Examples will be drawn primarily from applications of calculus, elementary number theory, and cryptography.

3.12.4 Outcome

Students will learn basic scripting and object-oriented programming, with the goal of being able to solve mathematical and scientific problems.

3.12.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Christine Haught	https://luc.box.com/s/qbtjm2odayq6hm0ko326ys2tsywdlag1

3.13 COMP 250: Introduction to Scientific and Technical Communication

This course trains students in writing clear, readable, and well-organized technical communications, including presentations, end-user documentation, internal project documentation, and scientific papers.

3.13.1 Credit Hours

3

3.13.2 Prerequisites

COMP 125: Visual Information Processing or *COMP 150: Introduction to Computing* or *COMP 170: Introduction to Object-Oriented Programming* or *COMP 215: Object-Oriented Programming with Mathematics*

3.13.3 Description

This course provides students with the knowledge and skills in writing, presentations, and other forms of technical communication expected of them in their future roles as science and technology professionals. The course covers the following topics: principles of organizing,developing, and writing technical information; forms and conventions common to scientific and technical disciplines; presentation of technical information to various audiences; principles and techniques of oral presentations; computer-aided visual presentation techniques. This course is officially approved for "writing-intensive" credit.

3.13.4 Outcome

Students will learn to write clear technical documentation.

3.13.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Roxanne Schwab	https://luc.box.com/s/25vqetcz9bj6f02cxl52quiehbruir3v
Spring 2012	Dr. Greenberg	https://luc.box.com/s/25vqetcz9bj6f02cxl52quiehbruir3v

3.14 COMP 251: Introduction to Database Systems

This course explores ways in which data collections are organized, stored, analyzed, and manipulated. Topics include relational databases, the SQL query language, and the basics of XML and web interfaces to data sets. Applications from a variety of domains illustrate the course's key concepts.

3.14.1 Credit Hours

3

3.14.2 Prerequisites

COMP 125: Visual Information Processing or *COMP 150: Introduction to Computing* or *COMP 170: Introduction to Object-Oriented Programming* or *COMP 215: Object-Oriented Programming with Mathematics*

3.14.3 Description

This course explores ways in which data collections are organized, stored, analyzed, and manipulated. Topics include relational databases, the SQL query language, and the basics of XML and web interfaces to data sets. Applications from a variety of domains illustrate the course's key concepts.

3.14.4 Outcome

Students will organize data in ways to emphasize relationships, write simple programs to process, visualize and graphically display data, mine data for patterns, and design web interfaces to data.

3.14.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Guy Bevente	https://luc.box.com/s/34jl29hua7f0vo4ul17hoe3rljv5lc1s

3.15 COMP 264: Introduction to Computer Systems

This course studies the hierarchy of abstractions and implementations that constitute a modern computer system, with a particular focus on issues of interest to programmers, typically including some systems programming instruction.

3.15.1 Credit Hours

3

3.15.2 Prerequisites

COMP 170: Introduction to Object-Oriented Programming or *COMP 215: Object-Oriented Programming with Mathematics*

3.15.3 Co-requisites

COMP 163: Discrete Structures or MATH 201: Elementary Number Theory (corequisite or prerequisite) or declared Computer Science, Computer Crime & Forensics or Information Technology Minor

3.15.4 Description

This course is designed to provide students with an understanding of the hierarchy of abstractions and implementations that comprise a modern computer system. The course is particularly geared towards topics of interest to a programmer, i.e., topics that affect the performance, correctness, or utility of user-level programs. Since this investigation is best

carried out using the C programming language, the course will include some instruction in C for programming familiar with Java.

3.15.5 Outcome

Understanding of system issues that affect the performance, correctness, or utility of user-level programs.

3.15.6 Syllabi

Semester/Year	Instructor	URL
Spring 2013	Dr. Greenberg	https://luc.box.com/s/9qnyyva3sgsm794zwddapj4ipm6nm4ol

3.16 COMP 266: Digital Electronics Laboratory

Also Offered As PHYS 266: Digital Electronics Laboratory

3.16.1 Credit Hours

3

3.16.2 Prerequisites

PHYS 112: College Physics II or permission.

3.16.3 Description

This course has lectures as well as labs. The topics that will be studied are; logic devices, oscillators and timers, microprocessor components, machine language and digital/analog conversion. Special emphasis would be put upon individual components of microprocessor systems.

3.16.4 Syllabi

No recent syllabi available.

3.17 COMP 271: Data Structures

This course introduces key data structures such as lists, sets, and maps, as well as their implementations. Performance and analysis of algorithms are covered along with applications in sorting and searching.

3.17.1 Credit Hours

3

3.17.2 Prerequisites

COMP 170: Introduction to Object-Oriented Programming or *COMP 215: Object-Oriented Programming with Mathematics*

COMP 163: Discrete Structures or MATH 201: Elementary Number Theory (corequisite)

3.17.3 Description

This continuation of *COMP 170: Introduction to Object-Oriented Programming* introduces the concepts of data abstraction and data structure, including stacks, queues, lists, sets, and trees. The issues of implementing a data structure in a language such as Java are examined using classes, arrays, and linked structures. Sorting and searching techniques are analyzed. The concepts of correctness and efficiency of algorithms are developed. Time/space comparisons of iterative algorithms with recursive algorithms are made. The course includes several major programming projects. A weekly lab component is required.

3.17.4 Outcome

Students will learn to design new data structures as well as learn to use existing data structures in applications.

3.17.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Putonti	https://luc.box.com/s/jzdj77yzwxime9xhtdbiz2exgja13xqi
Fall 2015	Dr. Sekharan	https://luc.box.com/s/jzdj77yzwxime9xhtdbiz2exgja13xqi

3.18 COMP 300: Data Warehousing and Data Mining

This course covers theory and practice of extremely large information storage (warehousing) and analysis (mining) mechanisms. With data growing at exponential rates knowledge gathering and exploration techniques are essential for gaining useful intelligence.

3.18.1 Credit Hours

3

3.18.2 Prerequisites

COMP 251: Introduction to Database Systems or *COMP 271: Data Structures*

3.18.3 Description

Data warehousing and data mining are two major areas of exploration for knowledge discovery in databases. These topics have gained great relevance especially in the 1990's and early 2000's with web data growing at an exponential rate. As more data is collected by businesses and scientific institutions alike,knowledge exploration techniques are needed to gain useful business intelligence. This course will cover a wide spectrum of industry standard tchniques using widely available database and tools packages for knowledge discovery.

Data mining is for relatively unstructured data for which more sophisticated techniques are needed. The course aims to cover powerful data mining techniques including clustering, association rules, and classification. It then teaches high volume data processing mechanisms by building warehouse schemas such as snowflake, and star. OLAP query retrieval techniques are also introduced.

3.18.4 Outcome

Students will be able to define and critically analyze data warehouse and mining approaches for fields such as security, forensics, privacy, and marketing.

3.18.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Channah Naiman	https://luc.box.com/s/vehaxaonmpiitl3ho3d02nnm85pglgmj

3.19 COMP 305: Database Administration

Business and scientific institutions increasingly use large commercial data base systems. This course teaches the theory and practice for the definition, security, backup, tuning, and recovery of these systems.

3.19.1 Credit Hours

3

3.19.2 Prerequisites

COMP 251: Introduction to Database Systems or *COMP 271: Data Structures*

3.19.3 Description

Businesses and scientific institutions have started using commercial grade database servers increasingly. Hence it has become important to install, configure, and manage the servers in an efficient manner to increase productivity. A number of tasks involved in maintaining a database server is quite different from say administering a computer system or a network. Deeper knowledge of the storage aspects of the server and how different server processes work are needed for successful administration of the server. Typically, the database administrator cares for security of data, backup and recovery, and space management. This course takes a user through the stages of setting up a database server environment, to design of a database, and finally tuning.

3.19.4 Outcome

Students will be able to use theory and pragmatic approaches to define and implement realistic solutions for large database administration environments.

3.19.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Dordal	https://luc.box.com/s/xtuvk5iqlqrchhe2y8u3rtu8z89xui0u

3.20 COMP 309: Numerical Methods

3.20.1 Aliases

MATH 309: Numerical Methods

3.20.2 Credit Hours

3

3.20.3 Prerequisites

MATH 215: Object Oriented Math Programming or *COMP 215: Object-Oriented Programming with Mathematics* or *COMP 170: Introduction to Object-Oriented Programming*

MATH 212: Linear Algebra

MATH 264: Ordinary Differential Equations

3.20.4 Course Description

Introduction to error analysis, numerical solution of equations, interpolation and approximation, numerical differentiation and integration, matrices and solution of systems of equations, numerical solution of ordinary and partial differential equations.

3.20.5 Syllabi

No recent syllabi available.

3.21 COMP 312: Free/Open Source Computing

This course will cover the fundamentals of Free and Open Source software development. Topics to be addressed include licensing, Linux, typical software development tools, applications, and techniques for managing remote servers.

3.21.1 Credit Hours

3

3.21.2 Prerequisites

COMP 251: Introduction to Database Systems or *COMP 271: Data Structures*

3.21.3 Description

This course will cover the fundamentals of Free and Open Source software development. Topics to be addressed include licensing, Linux, typical software development tools (e.g. compilers, scripting languages, build tools, and version control software), applications, and techniques for managing remote servers. Students will work on a significant developmentproject involving free and open-source software and learn how to participate in open-source projects effectively.

3.21.4 Outcome

Students will learn to implement projects involving Free and Open Source software and learn how to participate in open-source projects effectively.

3.21.5 Syllabi

Semester/Year	Instructor	URL
Summer 2014	George K. Thiruvathukal	https://luc.box.com/s/4vv84gz2qkoe773yfhpp8bign110x7fg
Spring 2014	Matt Butcher	https://luc.box.com/s/4vv84gz2qkoe773yfhpp8bign110x7fg

3.22 COMP 313: Intermediate Object-Oriented Development

Object-orientation continues to be a dominant approach to software development. This intermediate programming-intensive course studies the use of classes and objects with an emphasis on collaboration among objects.

3.22.1 Credit Hours

3

3.22.2 Prerequisites

COMP 271: Data Structures (strictly enforced)

3.22.3 Description

Object-orientation continues to be a dominant approach to software development. This intermediate programming-intensive course studies the use of classes and objects with an emphasis on collaboration among objects.

Overall Series of Object-Oriented Courses

- *COMP 170: Introduction to Object-Oriented Programming* (CS1) - simple objects representing scalars

- *COMP 271: Data Structures* (CS2) - collections of simple objects

- *COMP 313: Intermediate Object-Oriented Development* / *COMP 413: Intermediate Object-Oriented Development* - complex, interacting objects; basic design patterns

- *COMP 373: Objects, Frameworks, and Patterns* / *COMP 473: Object-Oriented Programming* - advanced design patterns and topics such as AOP(Aspect-Oriented programming)

COMP 313: Intermediate Object-Oriented Development / *COMP 413: Intermediate Object-Oriented Development* is also a prerequisite for other advanced software courses. Students interested in advanced software courses are encouraged to take *COMP 313: Intermediate Object-Oriented Development* / *COMP 413: Intermediate Object-Oriented Development* as soon as they have completed *COMP 271: Data Structures* so as to be eligible for these further courses.

Course Topics

- Data Structures of various types – linear vs. nonlinear, indexing vs. non-indexing, position vs. value-oriented

- Advanced Java, e.g. interfaces, annotations, exceptions, generics, collections, boxing/unboxing, array objects

- Object Modeling – UML, use cases and activity diagrams, class diagrams, archetypes, interaction diagrams

- Design by contract, interfaces, refactoring & generalization, design patterns (Adapter, Decorator, Composite, Strategy, Iterator, Abstract Factory, Visitor, . . .)

- Agile Development Process – evolutionary design, test-driven development, refactoring, . . .

- Tools – Eclipse, Subversion, JUnit, JMock, Ant, . . .

- Techniques – object pooling, garbage collection, performance profiling (NetBeans)

3.22.4 Outcome

A thorough understanding of the principles of object-orientation: abstraction, delegation, inheritance, and polymorphism; exposure to basic design patterns; programming experience in mainstream object-oriented languages such as C++ and Java.

You will take your software development abilities to the next level by building on your knowledge of data structures.

You will learn to design and implement more complex programs using good software engineering practices, including:

- Designing with interfaces and composition

- Design patterns

- Refactoring

- Test-driven development (TDD)

3.22.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Yacobellis	https://luc.box.com/s/i4f7shbe2nqv0mufd8sb7n9q6s1w0seq
Fall 2013	Dr. Läufer	https://luc.box.com/s/i4f7shbe2nqv0mufd8sb7n9q6s1w0seq

3.23 COMP 314/315: Problem Solving Strategies

This course allows students to sharpen problem-solving skills along with, or as part of, the ACM Programming Team. Groups generally work on old competition problems on alternate weekends, with short follow-ups during the next week.

3.23.1 Credit Hours

For 314: 1; For 315: 2

3.23.2 Prerequisites

For COMP 314:

COMP 271: Data Structures (or corequisite)

For COMP 315:

COMP 314: Problem Solving Strategies

3.23.3 Description

This course started as a means to give ACM programing team members academic credit for their participation. The class work will be designed to help these people. We will mostly work old programming team competition problems, alone and in groups, and discuss strategies for solving such problems effectively. You do not need to be interested in the official team to find these problems and the discussions about them interesting. You do NOT need to agree to be on the team to be in the class. You are welcomed to be in the class if you just want to sharpen your problem solving and triaging skills. The competition problems are reasonable ones to work on with or without the competition. They are are quite unlike typical course homework exercises: They come from all different areas, all different levels of difficulty, and the words describing them do not directly say any algorithm needed. A significant part of each problem is to classify it. Advanced students will certainly find challenging problems to work on. Students in Comp 271 should find enough to keep them productively occupied, and get a taste of more advanced techniques from class discussions.

If you *do* want to compete officially, see also http://anh.cs.luc.edu/314-315/prog-team.html.

By enrolling in the sequence 314-315 over two years, you satisfy 3 units of elective credit in Computer Science department majors.

In keeping with the programming competition format, we will arrange a schedule to meet about every two weekends for a 5 hour practice/problem solving session, with a weekday followup discussion for 1-1.5 hours on a weekday in the following week.

The organizational meeting to agree on practice and discussion times will be in the afternoon in the first week of classes. The time will be posted.

Contact Dr. Harrington if you cannot attend that session. Try to let him know ahead of time your available weekend times through October.

3.23.4 Outcome

Ability to work in small groups, quickly and accurately assessing and solving focused problems involving many sorts of programming knowledge.

3.23.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Harrington	https://luc.box.com/s/rsbewiqfemx7t6rwdhxezogkej1y0ao4

3.24 COMP 317: Social, Legal, and Ethical Issues in Computing

This course covers social, legal, and ethical issues commonly arising in key areas related to computing technologies.

3.24.1 Credit Hours

3

3.24.2 Prerequisites

None

3.24.3 Description

This course will explore a variety of ethical and legal issues facing those who use or program computers. Issues can be divided broadly into professional ethics, dealing with the ethical responsibilities of the programmer, and social issues, dealing with concerns we all have as citizens.

3.24.4 Outcome

Understanding of laws and issues in areas such as privacy, encryption, freedom of speech, copyrights and patents, computer crime, and computer/software reliability and safety; understanding of philosophical perspectives such as utilitarianism versus deontological ethics and basics of the U.S. legal system.

3.24.5 Recent Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Dordal	https://luc.box.com/s/so6ld4ie38wv8gpj363eqvcrh66rp0ud
Spring 2014	Benjamin Galatzer-Levy	https://luc.box.com/s/so6ld4ie38wv8gpj363eqvcrh66rp0ud

3.25 COMP 319: Introduction to Unix

An introduction to the UNIX operating system. Topics include files and directories, electronic mail, security, advanced file systems, network utilities, network file sharing, text utilities, shell programming, UNIX internals, UNIX system administration (essentials), the X windowing system, systems programming, and secure shell (SSH).

3.25.1 Credit Hours

1

3.25.2 Prerequisites

COMP 170: Introduction to Object-Oriented Programming

3.25.3 Description

An introduction to the UNIX operating system. Topics include files and directories, electronic mail, security, advanced file systems, network utilities, network file sharing, text utilities, shell programming, UNIX internals, UNIX system administration, the X windowing system, and systems programming. Programming assignments involve the UNIX shell script language.

3.25.4 Outcome

After taking this course, students will develop working knowledge of Unix and be able to use modern Unix operating systems such as Linux, OS X, or Solaris.

3.25.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Jose Luis Gahete	https://luc.box.com/s/5r3j8jq6qauotmt57osjwizszvgy1vpu

3.26 COMP 320: Software Systems Analysis

Software systems analysis and design document user needs, create system architecture, and guide implementation. This course teaches the Unified Modeling Language (UML), and uses current software tools for analysis and design.

3.26.1 Credit Hours

3

3.26.2 Prerequisites

COMP 163: Discrete Structures and *COMP 271: Data Structures*

3.26.3 Description

This course uses Unified Modeling Language notation to model the early software analysis and design phase. Object technology is critical to the understanding of the process of capturing business requirements and the course uses commercial software tools to perform systems analysis and design.

3.26.4 Outcome

Students will be able to use techniques of analysis and design, document results using UML, and understand how to communicate in team oriented settings.

3.26.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Conrad Weisert	https://luc.box.com/s/mfbpoi71269o3mts9jloehzhe38znp6q

3.27 COMP 322: Software Development for Wireless/Mobile Devices

3.27.1 Credit Hours

3

3.27.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

3.27.3 Description

This course will focus on the unique challenges, methods, tools, and technologies for developing software applications for wireless and mobile devices, such as personal digital assistants (PDA) and smart mobile phones. Topics include user interface design for smallscreen, multi-channel devices, programming techniques and memory management for devices with limited memory and processing power, data synchronization for mobile databases, and wireless network programming.

3.27.4 Syllabi

Semester/Year	Instructor	URL
Spring 2009	Dr. Honig	https://drive.google.com/file/d/0Bz_4VraMwHUoSGdaOU9JS0VUTms/edit?usp=sharing

3.28 COMP 324: Client-Side Web Development

3.28.1 Credit Hours

3

3.28.2 Prerequisites

COMP 271: Data Structures

3.28.3 Description

This course provides an in-depth study of the concepts and methods required for the design and implementation of the presentation layer of a web application. Topics include visual design, usability design, multi-channel and multi-modal applications, markup of static and dynamic content, content transformation, client-side executable content including client-side scripting and embedded applets, and web-based content management systems. Coursework includes several substantial programming projects (using technologies such as XHTML, XSTL, DHTML, JavaScript, PHP/ASP/JSP, Flash, and Zope/Plone).

3.28.4 Syllabi

Topic	Semester/Year	Instructor	URL
Client Side Web Design	Fall 2014	Dr. Albert	http://comp424.pacsites.org

3.29 COMP 328: Algebraic Coding Theory

3.29.1 Aliases

MATH 328: Algebraic Coding Theory

3.29.2 Credit Hours

3

3.29.3 Prerequisites

MATH 212: Linear Algebra

3.29.4 Description

Codes with algebraic structure for error control are examined. Block codes including Hamming codes and Reed-Muller codes, BCH codes, and other cyclic codes with algebraic structure and their implementation are treated. Other topics may include: convolutional codes, efficiency considerations, and Shannon's fundamental theorem of information theory.

3.29.5 Outcome

Students will learn both the theory and application of error-correcting codes.

3.29.6 Syllabi

No recent syllabi available.

3.30 COMP 330: Software Engineering

Students learn real-world theory and techniques organizations use to create high-quality software on time. Students work on a large programming team to create plans, review progress, measure quality, and make written and oral analyses of their project.

3.30.1 Credit Hours

3

3.30.2 Prerequisites

COMP 271: Data Structures

3.30.3 Description

Using an object-oriented language such as C++ or Java, the student (working in a small team) will learn to plan, design, implement, and test a large software project.

3.30.4 Outcome

Students will experience process based development, understand the dynamics of a professional software organization, and develop skills for implementing software with others.

3.30.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	George K. Thiruvathukal	https://luc.box.com/s/k7b024zws90bu8fjsogijpsg0t6qgsdq

3.31 COMP 331: Cryptography

This course introduces the formal foundations of cryptography and also investigates some well-known standards and protocols, including private and public key cryptosystems, hashing, digital signatures, RSA, DSS, PGP, and related topics.

3.31.1 Aliases

MATH 331: Cryptography

3.31.2 Prerequisites

Mathematical Preparation

COMP 163: Discrete Structures, MATH 313: Abstract Algebra, or MATH 201: Elementary Number Theory)

Programming Foundations

COMP 125: Visual Information Processing, *COMP 170: Introduction to Object-Oriented Programming*, or *COMP 215: Object-Oriented Programming with Mathematics* (or equivalent)

3.31.3 Description

This course introduces the formal foundations of cryptography and also investigates some well-known standards and protocols. The intended audience is senior undergraduate and beginning graduate students. The course will include topics selected from the following: information-theoretic security, private key encryption, DES, public key encryption, background on modular arithmetic, RSA, hashing and message authentication codes (MACs), digital signatures, DSS, key distribution and management, PGP, network security, and Fiat-Shamir protocol.

3.31.4 Outcome

Students will gain an understanding of cryptosystems widely used to protect data security on the internet, and be able to apply the ideas in new situations as needed.

3.31.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Stephen Doty	https://luc.box.com/s/crg2y0y7336tl8wvg6e8e3vxpufaxj97

3.32 COMP 333: Formal Methods in Software Engineering

As embedded and networked systems are becoming ever more ubiquitous, we depend increasingly on the correctness of the software that controls such systems. This course studies the formal specification, verification, and synthesis of software.

3.32.1 Credit Hours

3

3.32.2 Prerequisites

COMP 271: Data Structures

3.32.3 Description

This course covers formal methods used in the development of software. It studies languages for object-oriented modeling, such as the Unified Modeling Language (UML) along with its Object Constraint Language (OCL) and Action Semantics extensions; the specification of abstract data types, such as Z; and concurrency, such as process algebras and temporal logic.

3.32.4 Outcome

An understanding of the role of formal methods in the construction of software systems; proficiency in representative methods and tools, such as UML and ESC.

3.32.5 Syllabi

No recent syllabi available.

3.33 COMP 336: Markup Languages

This course is concerned with XML and its various component frameworks. The core frameworks to be covered include Document Object Model (DOM), Simple API for XML processing (SAX), the XML Path language (XPath), and XSLT.

3.33.1 Credit Hours

3

3.33.2 Prerequisites

COMP 251: Introduction to Database Systems or *COMP 271: Data Structures*

3.33.3 Description

This course covers Extensible Markup Language (XML) and its applications. This course will cover the core XML component frameworks, including XSLT (a transformational approach) and the various W3C specifications for manipulating XML documents programmatically, including the DOM and SAX frameworks. As well, this course will cover some advanced topics, including how to manage large XML documents and integration with databases. Please note that HTML will be occasionally used in this course; however, this course is not about HTML and students may want to acquire an HTML book and study it briefly before taking this course. This course is not about making cool web pages. The course has almost nothing to do with web pages and is focused more on modeling and the emergent notion of web services.

Students should expect the programming to be somewhat involved (intermediate to advanced). Most of the programming is based on straightforward data structures, such as trees, lists, and maps (collections found in Java). You are encouraged to study these topics on your own. Little time will be spent rehashing this preliminary knowledge.

3.33.4 Outcome

After taking this course, students will have working knowledge of XML and its connections to other ideas such as HTML, object models, relational databases, and network services.

3.33.5 Syllabus

Offering	Instructor	URL
Fall 2013	Matt Bone	https://luc.box.com/s/qferpymqukil4be7yw5yd7kclidq5r9e
Spring 2012	Dr. Thiruvathukal	http://markuplanguages.cs.courseclouds.com

3.34 COMP 337: Introduction to Concurrency

Many real-world software systems rely on concurrency for performance and modularity. This programming-intensive course covers analysis, design, implementation, and testing of concurrent software systems.

3.34.1 Credit Hours

3

3.34.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

3.34.3 Description

This course studies the architecture, design, and implementation of concurrent software systems. Process algebras, formal specification, and testing are used as tools in the engineering of concurrent systems; event-based programming frameworks and thread libraries are employed in the implementation of such systems. Coursework includes several substantial programming projects (in a language such as Java) involving applications of concurrency and event-driven programming such as graphical user interfaces and distributed services using Remote Method Invocation (RMI).

3.34.4 Outcome

An in-depth understanding of event-based and thread-based views of concurrency; the ability to develop concurrent software components using suitable languages, frameworks, and design patterns; familiarity with object-oriented modeling and development tools and test driven development.

3.34.5 Syllabi

No recent syllabi available.

3.35 COMP 338: Server-Based Software Development

Server-based web applications and services have become part of everyday life. This programming-intensive course covers analysis, design, implementation, and testing of multi-tiered server-based software systems along with typical tier-specific technologies.

3.35.1 Credit Hours

3

3.35.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

COMP 264: Introduction to Computer Systems (strongly recommended)

3.35.3 Description

This course studies the architecture, design, and implementation of multi-tiered server-based software systems. Each tier is studied along with the pertinent mechanisms: markup languages in the client tier, web application frameworks in the web tier, and server-side components in the business logic and persistence tiers. Pervasive issues such as integration, testing, security, and performance are discussed. Coursework includes several substantial programming projects (using a platform such as Java 2 Enterprise Edition).

3.35.4 Outcome

An understanding of software architecture and integration in the development of multi-tiered server-based software; familiarity with object-oriented modeling and development tools and test-driven development.

3.35.5 Syllabi

Semester/Year	Instructor	URL
Spring 2012	Dr. Läufer	https://luc.box.com/s/rqxwpqjrrmfumaf095x8g4p1qefbwss5

3.36 COMP 339: Distributed Systems

This course covers topics in modern distributed systems. This course places special emphasis on scalability (performance), reliability/fault tolerance, and security.

3.36.1 Credit Hours

3

3.36.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development or *COMP 374: Introduction to Operating Systems*

3.36.3 Description

This course is designed as a modern discussion of distributed computing systems, which represent one of the most important areas in academic and business computing today. Topics covered include distributed computing, interactive services, collaborative computing, and peer-to-peer sharing.

Various distributed frameworks and technologies will be explored, e.g. DNS, CORBA, Java RMI, SOAP (XML-based RPCs), and Globus.

3.36.4 Outcome

After taking this course, students should understand the essential ingredients of distributed systems and how to build distributed systems that are resilient to transient network failures and other potential anomalies.

3.36.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Dr. Thiruvathukal	https://luc.box.com/s/wtdgdnx6xiz6bwfhtejmo0u3nubhvln7

3.37 COMP 340: Computer Forensics

The course introduces the fundamentals of computer/network/internet forensics, analysis and investigations.

3.37.1 Credit Hours

3

3.37.2 Prerequisities

COMP 150: Introduction to Computing or *COMP 170: Introduction to Object-Oriented Programming* or *COMP 215: Object-Oriented Programming with Mathematics*/MATH 215: Object Oriented Math Programming

COMP 264: Introduction to Computer Systems or *COMP 317: Social, Legal, and Ethical Issues in Computing* or *COMP 343: Introduction to Computer Networks*

3.37.3 Description

The course introduces the fundamentals of computer/network/internet forensics, analysis and investigations.

3.37.4 Outcome

The student will learn Computer Software and hardware relevant for analysis and investigative and evidence-gathering protocols.

3.37.5 Syllabi

Semester/Year	Instructor	URL
Fall 2014	Tom Yarrish	https://luc.box.com/s/cuczlxoabjg5fpat09u17g7ae2q06kn2

3.38 COMP 341: Human-Computer Interaction

3.38.1 Credit Hours

3

3.38.2 Prerequisites

COMP 271: Data Structures

3.38.3 Description

This course studies the interaction between humans and computer-based systems. The course will provide students with the methods for evaluating, designing, and developing better interfaces between humans and systems. Students will acquire an awareness of different design and evaluation methods as well as practical, effective, and cost-conscience methods for improving systems and their interfaces.

3.38.4 Syllabi

Semester/Year	Instructor	URL
Spring 2013	Dr. Albert	http://mva.me/edu/hci/

3.39 COMP 343: Introduction to Computer Networks

This course surveys packet-switched computer networks and attendant communication protocols, using the TCP/IP protocol suite on which the Internet is based as the primary model. Some Java programming is required.

3.39.1 Credit Hours

3

3.39.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

3.39.3 Description

This course surveys the construction of modern packet-switched networks, focusing on TCP/IP. At the LAN layer, protocols such as Ethernet and Wi-Fi are described. At the IP layer, a variety of routing-update protocols are studied. At the Transport layer, the course addresses the creation of end-to-end reliability and the management of network-wide congestion. The course also introduces basic issues in network security and in real-time communications. Some programming, generally in a language like Java or C#, may be required.

3.39.4 Outcome

Students will understand how the Internet is constructed, how data is routed to its destination, how connections are made, how congestion is handled, and how security can be addressed.

3.39.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Dordal	https://luc.box.com/s/4kwe8ihlx30yr0me1zlqv7tbvtspijvd

3.40 COMP 344: Web Services Programming

3.40.1 Credit Hours

3

3.40.2 Prerequisites

- *COMP 313: Intermediate Object-Oriented Development*
- instructor permission if prerequisite is missing

3.40.3 Description

Web services are building blocks for enterprise applications that use open data exchange standards and transport protocols to exchange data with calling clients. This course studies the architectures, frameworks, and tools required to develop and compose web services and clients, as well as integrate service-oriented systems with legacy systems.

3.40.4 Syllabus

Offering	Instructor	URL
Fall 2012	Dr. Laufer	http://laufer.cs.luc.edu/teaching/433

3.41 COMP 346: Introduction to Telecommunications

This course introduces the fundamental concepts of telecommunication networks, including requirements of voice networks, analog versus digital transmission, data link protocols, SONET, ATM, cellular phone systems, and the architecture of the current telephone system.

3.41.1 Credit Hours

3

3.41.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

3.41.3 Description

This course introduces the fundamental concepts of telecommunication networks. Underlying engineering principles of telephone networks, computer networks and integrated digital networks are discussed. Topics in the course include: telephone and data networks overview; OSI layers; data link protocol; flow control, congestion control, routing; and local area networks (Ethernet, Token Ring and FpI).

3.41.4 Outcome

Students will understand how modern telephone systems work.

3.41.5 Syllabi

Semester/Year	Instructor	URL
Spring 2013	Dr. Dordal	https://luc.box.com/s/ggz236rzzkm8j93vwa8cwx4ukvjqw2lh

3.42 COMP 347: Intrusion Detection and Computer Security

This course covers techniques and algorithms for detecting unusual usage patterns that typically signal a break-in, including techniques for detecting evasive or stealthy attacks. Also covered are differences in detecting local versus network intruders. Additional topics: computer viruses, computer security management, computer forensics.

3.42.1 Credit Hours

3

3.42.2 Prerequisites

COMP 150: Introduction to Computing or *COMP 170: Introduction to Object-Oriented Programming*

3.42.3 Description

This course will cover techniques for detecting the unusual usage patterns that typically signal a break-in. The course will also consider differences in detection of local intruders versus intrusion over networks. Finally issues in the prosecution of those breaking in to computers, particularly evidentiary issues are explored.

3.42.4 Outcome

Students will learn to configure ID systems (e.g. SNORT) and analyze their output. They will also understand both network-based and host-based monitoring techniques.

3.42.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Corby Schmitz	https://luc.box.com/s/57n40325p8xudttu7i35nxfb3tl9ieh5

3.43 COMP 348: Network Security

This course continues some of the topics introduced in COMP 347. The course covers methods and tactics to keep network attackers at bay and teaches mechanisms to identify and potentially stop potential intruders. Analyses of specific attack mechanisms may be considered.

3.43.1 Credit Hours

3

3.43.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

3.43.3 Description

This course will involve a discussion of the methods and tactics used to keep attackers at bay as well as the mechanisms by which we can identify and potentially stop potential "bad guys." It will involve the following topics as they all relate to the overall security posture which makes computing safer: Encryption, authentication, firewalls, NAT/PAT, restricted access policies, intrusion detection and other security frameworks. The goal is to gain an understanding of how to secure computers and computing environments. Instructor permission is based on relevant industry experience with networking including TCP/IP.

3.43.4 Outcome

An understanding of how to secure networks using encryption, authentication, firewalls, NAT/PAT, restricted access policies, intrusion detection and other security frameworks.

3.43.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Corby Schmitz	https://luc.box.com/s/gwem0sfi31naca0halwbvk0xyj5ww6en

3.44 COMP 349: Wireless Networks and Security

This course will explore the wireless standards, authentication issues, and common configuration models for commercial versus institutional installations and analyze the security concerns associated with this ad-hoc method of networking.

3.44.1 Credit Hours

3

3.44.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

3.44.3 Description

In a mobile world, the ability to gain network access in a convenient manner, but yet securely, is becoming more and more of a requirement. This course will explore the wireless standards, authentication issues, common configuration models for commercial versus institution installs and analyze the security concerns associated with ad-hoc and standards-based methods of networking.

3.44.4 Outcome

Students will gain an understanding of wireless networking, protocols, and standards and security issues.

3.44.5 Syllabi

Semester/Year	Instructor	URL
Summer 2015	Corby Schmitz	https://luc.box.com/s/cl40id5nqxyqno18tj5z9ne9knkw9ypk

3.45 COMP 351: Network Management

This course introduces the current state of the art in automated management of computer networks, including protocols such as SNMP and its attendant naming conventions, network management systems, and important issues in administrative network configuration.

3.45.1 Credit Hours

3

3.45.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

3.45.3 Description

Networks today are high-speed, heterogeneous, large-scale and delivers different media including data, audio and video. How do you effectively manage today's complex computer networks? This class provides complete yet accessible answers to network managers and researchers in this field. The course covers the basics of network management, alternative architectures, evaluation techniques, network management system components, SNMP and CMIP management protocols and the ISO network management applications: fault management, performance management, configuration management, security management, and accounting management. The course emphasizes the practical experience of developing network and distributed systems management tools using the SNMP++ and AdventNet wrappers. This course also highlights the latest advances in networks and distributed management area and shows case studies of academic and industrial systems such as HiFi, SMARRT, OpenView, NetView and Tivoli.

Class meets with INFS 793.

3.45.4 Outcome

Students will become familiar with the SNMP protocol, with how large-scale Network Management Systems operate and are configured, and with advanced network configuration.

3.45.5 Syllabi

Semester/Year	Instructor	URL
Spring 2011	Dr. Dordal	https://luc.box.com/s/qh7vz7emcb2bcou3c2v7nky4ghhwet68

3.46 COMP 353: Database Programming

This course introduces relational and object databases to support database creation and application development. Use of commercial database products will give a practical orientation.

3.46.1 Credit Hours

3

3.46.2 Prerequisites

COMP 251: Introduction to Database Systems or *COMP 271: Data Structures*

3.46.3 Description

This course covers the fundamentals of database application development using C++, C, or Java by accessing a transaction-oriented database server. A commercial database environment such as Oracle is used. Apitional topics may include enabling access to database via the web, and administering large databases.

3.46.4 Outcome

Students will learn SQL, database design and application development using the latest software tools. Students will also learn techniques for web based data retrieval and manipulation.

3.46.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Sekharan	https://luc.box.com/s/2oy46wkn3wvkodrqp7lzn8tckcmzbcjo

3.47 COMP 356: Computer Principles of Modeling & Simulation

3.47.1 Credit Hours

3

3.47.2 Prerequisites

MATH 215: Object Oriented Math Programming or *COMP 215: Object-Oriented Programming with Mathematics* or *COMP 170: Introduction to Object-Oriented Programming*

MATH 132: Applied Calculus II or MATH 162: Calculus II

STAT 203: Statistics

3.47.3 Description

The tool of simulation is probably the most useful tool for analyzing problems which arise in real applications which are much too complicated to be analyzed mathematically. One example is the study of inventory control policies. When should the inventory be checked? How many units should be ordered? What are the anticipated consequences of these decisions? We simulate the random character of the demand for the product and the delay in receiving orders of the product. The objective is to provide management with relevant information for constructing an acceptable inventory policy.

3.47.4 Syllabi

No recent syllabi available.

3.48 COMP 362: Computer Architecture

This course covers computer design from the level of digital logic and circuit design to high-level computer organization.

3.48.1 Credit Hours

3

3.48.2 Prerequisites

COMP 264: Introduction to Computer Systems or Comp 360 or comparable background, including but not limited to the following:

- Understanding of basic computer organization, including familiarity with such components as CPU, ALU, multiplexors, registers, main memory, caches, and buses
- familiarity with the roles of compilers, assemblers, and operating systems
- some familiarity with assembly language
- ability to understand simple C programs and to run programs in a UNIX environment, and
- familiarity with the representation of numbers in digital computers

3.48.3 Description

This course presents key principles underlying the design of modern digital computers. The course introduces quantitative techniques used to guide the design process. It describes CPU performance issues and introduces instruction set architectures. The course then uses a hypothetical computer design, with a simple RISC architecture, to show how modern digital computers are implemented, first using a simple non-pipelined implementation, followed by a higher-performace pipelined implementation. The major hazards introduced by pipelining, including structural hazards, data hazards, and control hazards are discussed and techniques for overcoming them are described. Additional topics covered in this course include the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

3.48.4 Outcome

A basic understanding of how computers work at many levels and how to use various analytical tools and techniques to design computer components.

3.48.5 Syllabi

No recent syllabi available.

3.49 COMP 363: Design and Analysis of Computer Algorithms

Theoretical design and analysis of computer algorithms may be supplemented by small amounts of programming.

3.49.1 Credit Hours

3

3.49.2 Prerequisites

- *COMP 163: Discrete Structures* or MATH 201: Elementary Number Theory
- *COMP 271: Data Structures*
- MATH 131: Applied Calculus I or MATH 161: Calculus I

3.49.3 Description

The central goal of this course is for students to gain basic skills in designing and implementing efficient and effective computer programs. The course begins by developing models and mathematical tools for measuring the efficiency of algorithms. Then students are introduced to a variety of useful data structures and to algorithms for a variety of fundamental problems. Finally, the course will provide an introduction to classification of computational problems into different complexity classes. The course will include a small amount of actual programming in addition to theoretical analysis.

3.49.4 Outcome

The ability to design and analyze efficient algorithms; understanding of the necessary models and mathematical tools; understanding of a variety of useful data structures and fundamental algorithms; exposure to the classification of computational problems into different complexity classes.

3.49.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Dr. Albert	https://luc.box.com/s/r5s1mpy8vt0bv2y4y9d87ls0bgtlnloe
Spring 2013	Dr. Harrington	https://luc.box.com/s/r5s1mpy8vt0bv2y4y9d87ls0bgtlnloe

3.50 COMP 364: High-Performance Computing

This course covers parallel architectures and parallel models of computation. Algorithms for achieving high performance in various computational contexts are discussed. Models such as shared memory, message passing, and hybrid modes of computing are introduced.

3.50.1 Credit Hours

3

3.50.2 Prerequisites

COMP 264: Introduction to Computer Systems and *COMP 271: Data Structures*

3.50.3 Description

This course is motivated by the realization that cpu speeds are levelling off at between 3GHz and 5GHz. The need for programs to be faster and more efficient is all the more important at a time when applications are getting increasingly larger and more complex. We will introduce techniques to gain performance boost in Java programs and C++ (or C) programs by discussing the use of multiple processors.

This course makes use of departmental HPC resources. Please see the Systems Handbook at http://syshandbook.cs. luc.edu for details about our various HPC computing systems and partner resources.

3.50.4 Outcome

Students will learn how to engineer solutions to practical problems in multiprocessor architectures and using large physical memories.

3.50.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Christopher P Stone	https://luc.box.com/s/rjmodg9txmk6nzx5lqxcvfwyql5iim36

3.51 COMP 366: Microcomputer Design and Interfacing

3.51.1 Cross-Listing

Also Offered PHYS 366

3.51.2 Credit Hours

3

3.51.3 Prerequisites

COMP 264: Introduction to Computer Systems or PHYS 266

3.51.4 Description

This course includes lecture and laboratory work. Topics that the course will hit on are; Microprocessor logic, instruction, and clocked sequential circuits; memory devices; data acquisition, manipulation and transfer circuitry; machine and assembly languages; microprocessor support devices.

3.51.5 Syllabi

Semester/Year	Instructor	URL
Fall 2014	William Honig	https://luc.box.com/s/1ashm2fvtws6jni8ewld23jusvpcmokv

3.52 COMP 370: Software Quality, Metrics, and Testing

The course teaches software testing and quality control concepts, principles, and techniques including black box and white box testing, coverage testing, test case development, and regression testing.

3.52.1 Credit Hours

3

3.52.2 Prerequisites

COMP 271: Data Structures

3.52.3 Description

In this programming intensive course, students will learn effective automation, testing, and use of software metrics through the practices of Test Driven Development and Continuous Deployment. The first half of the course will focus on the fundamentals of Test Driven Development and test automation. The second half of the course will build upon those skills and allow the students to create acceptance test and deployment automation tools. Finally, students will learn how to measure user feedback effectively, thereby allowing them to safely deploy changes to running production software multiple times per day, rather than once every few weeks or months.

After taking this course, students should be able to:

- Employ Test Driven Development to create automated unit tests and improve the design of software.
- Create tools for build, test, and deployment automation.
- Understand the benefits and costs of automated acceptance testing.
- Use automated tests and metrics to address cross-cutting concerns such as performance and scalability.
- Possess the basic skills necessary to create, test, deploy, and maintain an application written for a cloud computing environment.

3.52.4 Outcome

Students will learn how to prevent errors, how to get 'bugs' out of software, and be able to apply this knowledge in other courses and projects.

3.52.5 Syllabi

Semester/Year	Instructor	URL
Fall 2014	Conrad Weisert	https://luc.box.com/s/f5q8r3a9h7opfcwcy0fraoqvri6ev79m

3.53 COMP 372: Programming Languages

There are over two thousand programming languages. This course studies several languages that represent the much smaller number of underlying principles and paradigms.

3.53.1 Credit Hours

3

3.53.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

3.53.3 Description

There are over two thousand programming languages. This course studies several languages that represent the much smaller number of underlying principles and paradigms.

3.53.4 Outcome

An understanding of key principles and paradigms underlying the design and implementation of commonly used programming languages; exposure to formal mechanisms for describing language syntax and semantics; programming experience in several representative languages.

3.53.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Läufer	https://luc.box.com/s/xkf3pxrt6tqoddj75337j55c96w80zml

3.54 COMP 373: Objects, Frameworks, and Patterns

Object-orientation continues to be a dominant approach to software development. This advanced programming-intensive course studies object-oriented analysis, design, and implementation from a design patterns perspective.

3.54.1 Credit Hours

3

3.54.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

You need to have knowledge of Java, C# or a similar object-oriented language, data structures, and fair sophistication with OOP - some experience with designing interacting, cooperating classes that would come in an intermediate discussion of software patterns. Software engineering is recommended but not required like the background listed above.

3.54.3 Description

Object-oriented programming has become the dominant software development paradigm of the 1990s. This course assumes familiarity with the main constituents of the object-oriented methodology:

- OOP = abstraction + inheritance + polymorphism

Building on this foundation, the course studies three major topics:

Design Patterns

First, the course discusses patterns, reusable solutions to recurring software design problems. Common design patterns such as Composite, Decorator, and Command typically involve several classes related by composition and inheritance.

Implementation

Next, the course examines how object-oriented language features such as composition, encapsulation, inheritance, and dynamic binding can be implemented. These features are studied in the context of an interpreter for a simplified object-oriented language.

Components

Finally, the course explores programming with reusable components. Specifically, the standard features of component architectures are discussed: properties, events, methods, and persistence, and introspection.

Other Information

This course is heavily project-oriented. Students study each major topic in the context of a substantial programming project. The reuse of existing class libraries and software components is emphasized.

Grading will be based upon the following components: programming projects, quizzes, in-class exams, and the final exam.

3.54.4 Outcome

Proficiency in the use of object-oriented languages, frameworks, and patterns; advanced understanding of key language mechanisms such as delegation, inheritance, polymorphism, and reflection; familiarity with object-oriented modeling and development tools and test-driven development.

3.54.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Berhane Zewdie	https://luc.box.com/s/9563fmryda6kwsti73ipeol1nypj0v6o

3.55 COMP 374: Introduction to Operating Systems

This course introduces principles of operating systems and how they are designed. Various important parts of operating systems such as memory addressing, file structures, processes, and threads are covered.

3.55.1 Credit Hours

3

3.55.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

3.55.3 Description

This is an introductory course in Operating Systems discussing both standalone and distributed environments. The focus of the course is to understand the underlying technologies that make contemporary operating systems work efficiently. We will discuss processes, threads, synchronization, I/O, file systems, memory management, transactions and system coordination techniques.

Through this course we will discover how these technologies are integrated into the systems we use today and then utilize these technologies and apply them to practical applications. This is NOT a programming intensive course, however, students will be expected to complete some programming in C with plenty of examples and assistance along the way. You certainly don't need to know how to program in C today. In addition, the completion of a technical paper on an OS related subject will also be expected.

Everyone currently in, or planning to enter the IT field should have a grasp on these components as they effect every area of the day to day operation of IT technology. Reference systems will include both Linux and Windows.

3.55.4 Outcome

Students will learn the different parts of an operating system at a functional level and how they interact with each other.

3.55.5 Syllabi

Semester/Year	Instructor	URL
Spring 2013	Joe Kaylor	https://luc.box.com/s/6do6ewznl4fsppy9ofqrsoqm74t00bwz
Spring 2012	Joe Kaylor and George K. Thiruvathukal	https://luc.box.com/s/6do6ewznl4fsppy9ofqrsoqm74t00bwz

3.56 COMP 376: Formal Languages and Automata

This course introduces formal language theory, including such topics as finite automata and regular expressions, pushdown automata and context-free grammars, Turing machines, undecidability, and the halting problem.

3.56.1 Credit Hours

3

3.56.2 Prerequisites

COMP 163: Discrete Structures or MATH 201: Elementary Number Theory or MATH 212: Linear Algebra

3.56.3 Description

This course will study three mutually related topics: languages, machines, and computability. The mathematical ideas developed in this course are useful in many areas of computer science, including the design and specification of programming languages, construction of compilers, and exploring the capabilities and limitations of mechanical computation. This subject is important for the scientific foundations it lays for computer science, for the philosophical concerns it raises about the nature of computation, and for the sheer elegance it brings in to the studies related to a variety of applications. Some of the most fundamental discoveries in computer science identify connections among languages, machines, and computability. Furthermore, some of the most challenging questions at the heart of computer science also arise from these topics. The course will cover a majority of the following topics: regular languages, finite automata, determinism and nondeterminism in finite automata, applications to searching and pattern matching, context-free languages, push-down automata, applications to compiler design, computability theory, Church-Turing thesis, Turing machines, undecidability, recursive and recursively enumerable languages, reductions among languages, resource-bounded computation, Kolmogorov complexity.

3.56.4 Outcome

An understanding of the theoretical underpinnings of computability and complexity in computer science.

3.56.5 Syllabi

Semester/Year	Instructor	URL
Fall 2014	Stephen Doty	https://luc.box.com/s/2idf1nl0e8wxwwy4gjysvc01aumi23el

3.57 COMP 377: IT Project Management

3.57.1 Credit Hours

3

3.57.2 Prerequisites

COMP 251: Introduction to Database Systems or *COMP 271: Data Structures*

3.57.3 Description

This course is an introduction to the philosophy and practice of project management. The course involves a student group project to investigate and plan a 'real world' IT project that specifies project objectives, schedules, work breakdown structure and responsibilities, a written interim report, and a final oral and written report.

3.57.4 Outcome

Students will learn time management, work-flow management, and team dynamics to design, implement and test large-scale software projects.

3.57.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Conrad Weisert	https://luc.box.com/s/rkt3e3ifk353jrese0b87fpmr4397xew

3.58 COMP 378: Artificial Intelligence

This course introduces artificial intelligence theory and programming.

3.58.1 Credit Hours

3

3.58.2 Prerequisites

COMP 271: Data Structures

3.58.3 Description

Concepts of problem representation and search, knowledge representation and reasoning. Selected topics may include game playing, theorem proving, natural language processing, machine learning, connectionist models, expert systems, robotics, pattern recognition, machine vision, neural networks.

3.58.4 Outcome

Student will learn basic theory of artificial intelligence and be able to build small applications based on it.

3.58.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Conrad Weisert	https://luc.box.com/s/k4u8xook39g0nnw0seuhixvt02eyro1k

3.59 COMP 379: Machine Learning

3.59.1 Credit Hours

3

3.59.2 Prerequisites

COMP 271: Data Structures

3.59.3 Description

Machine learning is the process of making predictions and decisions from data without being explicitly programmed. Topics include a variety of supervised learning methods. Ensemble approaches are used to combine independent models efficiently. Unsupervised and semi-supervised methods demonstrate the power of learning from data without an explicit training goal.

3.59.4 Outcome

Students in this course will learn how to apply sophisticated algorithms to large data sets to make inferences for prediction or decision making.

3.59.5 Syllabi

Topic	Semester/Year	Instructor	URL
Machine Learning	Fall 2015	Dr. Albert	http://machinelearning.pacsites.org

3.60 COMP 380: Introduction to Computer Graphics

This course introduces modern theory and practices in 3-D computer graphics, stressing real-time interactive applications using libraries like OpenGL.

3.60.1 Credit Hours

3

3.60.2 Prerequisites

COMP 271: Data Structures

The requirements for the course are some familiarity with C or C++, basic linear algebra (matrices and vector geometry), and differential calculus.

3.60.3 Description

Computer graphics is the art and science of producing images using a computer. We will study the theory of computer graphics, in particular 3D rendering, while gaining a working knowledge of the OpenGL graphics library. Much of the computer graphics we see in the media was created using software based on the Open GL graphics library, which has quickly become an industry standard.

A word of warning-computer graphics makes considerable use of mathematics-elementary linear algebra, vector analysis and differential calculus.

Other Information

There will be 7-8 programming assignments, a midterm, and a final. You will probably do your assignments using Visual C++ under either Windows 95 or Windows NT.

3.60.4 Outcome

Student will learn how to program real-time interactive applications using libraries like OpenGL.

3.60.5 Syllabi

No recent syllabi available.

3.61 COMP 381: Bioinformatics

Students will engage in the applications of computer-based tools and database searching to better understand DNA and protein structure, function, and evolution.

3.61.1 Credit Hours

3

3.61.2 Prerequisites

BIOL 101: General Biology I (or equivalent) or NTSC 104

3.61.3 Description

This course is cross-listed with Biology. Students will engage in the applications of computer-based tools and database searching to better understand DNA and protein structure, function, and evolution.

3.61.4 Outcome

Students will be able to apply their understanding of genetic and evolutionary processes to the appropriate use of computer software and manipulation of large databases to accurately predict structural, informational, functional, and evolutionary characteristics of DNA and protein sequences.

3.61.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Dr. Putonti	https://luc.box.com/s/y07dk2vx4bdg84fr8zmqu758msy93e3q

3.62 COMP 382: Introduction to Compilers

This course covers the basics of writing a compiler to translate from a simple high-level language to machine code. Topics include lexical analysis, top-down and LR parsing, syntax-directed translation, and code generation and optimization. Students will write a small compiler.

3.62.1 Credit Hours

3

3.62.2 Prerequisites

COMP 264: Introduction to Computer Systems and *COMP 313: Intermediate Object-Oriented Development*

3.62.3 Description

This course covers the basics of writing a compiler to translate from a simple high-level language to machine code. Topics include lexical analysis, top-down and LR parsing, syntax-directed translation, and code generation and optimization. Students will write a small compiler.

3.62.4 Outcome

Students will learn how a compiler is built.

3.62.5 Syllabi

No recent syllabi available.

3.63 COMP 383: Computational Biology

This course presents an algorithmic focus to problems in computational biology. It is built on earlier courses on algorithms and bioinformatics. Problems and solutions covered in this course include gene hunting, sequence comparison, multiple alignment, gene prediction, trees and sequences, databases, and rapid sequence analysis.

3.63.1 Credit Hours

3

3.63.2 Prerequisites

COMP 381: Bioinformatics

COMP 271: Data Structures

COMP 163: Discrete Structures

BIOL 388: Bioinformatics

3.63.3 Description

This course presents an algorithmic focus to problems in computational biology. As such it is built on earlier courses on algorithms and bioinformatics. Problems and solutions covered in this course include Gene Hunting, Sequence Comparison, Multiple Alignment, Gene Prediction, Trees and Sequences, Databases and Rapid sequence analysis.

3.63.4 Outcome

Students will learn, in detail, foundational methods and algorithms in bioinformatics.

3.63.5 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Heather E. Wheeler	https://luc.box.com/s/ktamvof6a5lbzmcvaszsunrddvm5t8nt

3.64 COMP 384: Metagenomics

Exploration of next-generation sequencing technologies for assessing microbial diversity in ecological niches.

3.64.1 Credit Hours

3

3.64.2 Prerequisites

BIOL 282 and Instructor Consent

3.64.3 Description

This course is cross-listed with Biology.

Exploration of next-generation sequencing technologies for assessing microbial diversity in ecological niches.

3.64.4 Outcome

Students will gain hands-on experience with metagenomic methodologies while working in an interdisciplinary, collaborative setting.

3.64.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Putonti	https://luc.box.com/s/xaz9hhvvql03to5o8g6xpj3tny2eia08

3.65 COMP 386: Computational Neuroscience

Introduces computational methods to understand neural processing in the brain.

3.65.1 Credit Hours

3

3.65.2 Prerequisites

COMP 150 or COMP 170 or Instructor Consent

3.65.3 Description

Introduces computational methods to understand neural processing in the brain. Levels of representation from low-level, temporally precise neural circuits to systems-level rate-encoded models, to information-theoretic approaches. Emphasis on sensory systems, primarily vision and audition, most readily demonstrating the need for such computational techniques.

3.65.4 Outcome

Appreciation that many aspects of neuroscience cannot be understood without appropriate mathematical or computational frameworks, and ability to adeptly apply these frameworks in the various domains of neuroscience.

3.65.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr.Albert	https://docs.google.com/document/d/ 1eUdWoTmRYZtFVt0Sgnm4f2wnXKpRGmR1stZE9YpcYsQ/edit?usp=sharing

3.66 COMP 388: Topics in Computer Science

3.66.1 Credit Hours

3

3.66.2 Prerequisites

Department and instructor consent

3.66.3 Description

This course is used to introduce emerging topics in computer science that do not yet have a regular course number. Content of the course varies. Recent topics have included

- Enterprise Software Development
- Rapid Application Development Metholodgy (.NET)
- Robotics Software Development
- System Standards and Requirements
- Topics in Algorithms

3.66.4 Outcome

Understand an emerging area of Computer Science.

3.66.5 Syllabi

Topic	Semester/Year	Instructor	URL
Compressed Introduction to Computer Science	Spring 2016	Andrew N. Harrington	https://luc.box.com/s/ 8jcnv6guf5sdtuxwh0tqralcicigqebu
Physical Design and Fabrication	Spring 2016	Jonathan Durston	https://luc.box.com/s/ meu8ky97rnknvq7ya0x05bsxuovueqiq
Data Structures	Fall 2015	Peter L. Dordal	https://luc.box.com/s/ 7dyol8ww3zc6ntebbgm5homhp10r7ze5
Web Services	Fall 2015	Berhane Zewdie	https://luc.box.com/s/ dbuowyp8464unfb0iaoe2buizb7unqy9
I.T. Project Management	Spring 2015	Conrad Weisert	https://luc.box.com/s/ sbrd7k2e2g3zozqcndpmcgtdreq0f92i
Past Offerings			https://luc.box.com/s/ npdm0gbxtao6ahenwnddk462745q5tpj

3.67 COMP 388: Foundations of Computer Science 1

3.67.1 Credit Hours

3

3.67.2 Prerequisites

Comp 150 or Comp 163 or some programming experience or comfort with math at the level of pre-calculus; restricted to students with a bachelor degree and to advanced transfer students, with departmental permission

3.67.3 Description

This compressed course offered in the first half of the semester is aimed at mature students who are interested in a quick start in computer science, like those entering the MS programs in Computer Science, Software Engineering, or Information Technology, who are lacking the prerequisite knowledge from introductory computer science course, *COMP 170: Introduction to Object-Oriented Programming*, which is covered in this course. This course is intended to be followed by *COMP 388: Foundations of Computer Science 2* in the second half of the semester, covering the content of *COMP 271: Data Structures*.

3.67.4 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Harrington	http://anh.cs.luc.edu/388Intro/index.html

3.68 COMP 388: Foundations of Computer Science 2

3.68.1 Credit Hours

3

3.68.2 Prerequisites

COMP 170: Introduction to Object-Oriented Programming or *COMP 388: Foundations of Computer Science 1*; restricted to students with a bachelor degree or to advanced transfer students, with departmental permission

3.68.3 Description

This course in the second half of the semester is the compressed version of *COMP 271: Data Structures*. The course is generally preceded by *COMP 388: Foundations of Computer Science 1*, the compressed introduction to computer science. *COMP 388: Foundations of Computer Science 2* is aimed at mature students who are interested in continuing a quick start in computer science, like those entering the MS programs in Computer Science, Software Engineering, or Information Technology, who are lacking the prerequisite knowledge from *COMP 271: Data Structures*.

3.68.4 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Dordal	http://webpages.cs.luc.edu/~pld/courses/388/fall14/

3.69 COMP 390: Broadening Participation in STEM (Computing, Math & Science)

Students will learn about underrepresentation of various population groups in STEM fields (science, technology, engineering, mathematics), as well as some of the reasons and negative effects of this situation. They will learn about techniques and educational materials for ameliorating this situation and will engage in relevant service learning activities.

3.69.1 Credit Hours

3

3.69.2 Prerequisites

None

3.69.3 Description

Students will learn about underrepresentation of various population groups in STEM fields (science, technology, engineering, mathematics), as well as some of the reasons and negative effects of this situation. They will learn about techniques and educational materials for ameliorating this situation and will engage in relevant service learning activities.

3.69.4 Outcome

Students gain first-hand experience with broadening STEM participation and seeing how they can make a difference in the lives of other students and contribute to national needs.

3.69.5 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Greenberg	https://luc.box.com/s/vn8awlg9aod3iyesmsh0pdrwdtqbecc2

3.70 COMP 391: Internship in Computer Science

Students work outside the classroom applying and extending their computer science skills, typically for at least 150 hours for 3 credits. A memorandum of understanding is required between a student, his or her employer, and the Undergraduate Program Director, followed by final reports from the student and the employer.

3.70.1 Credits

Variable, 1-6

Undergraduate Internship Coordinator approval

3.70.2 Description

An opportunity to obtain experience in software development, design, or related activities in computer science in a professional setting.

Selection for Internships in Computer Science:

1. Students interested in internship must first find an employer/supervisor willing to provide them with a job. Internships are most advantageous to students if they already have considerable experience in upper level Computer Science courses. Skills from upper level courses are also going to help students be hired.

2. The student must make a formal request to the Undergraduate Internship Coordinator, supplying a job Description. The Undergraduate Internship Coordinator is looking for positions where the student will use and extend knowledge from upper level courses.

3. The Undergraduate Internship Coordinator reviews the student's performance in general and in relevant Comp classes, and either approves the request or suggests what further steps the student should make before securing an internship.

4. By the start of the internship the student must deliver the Internship Memorandum of Understanding (see below) signed by both the supervisor, student, and Undergraduate Internship Coordinator, outlining the expectations of all parties.

Presently the standard expectation is 50 hours of work relevant to a computer science education for each unit of credit. Hence 150 hours are expected for 3 credits. Work may overlap terms, and then credit is offered for the term in which the internship is completed. For more formal details, please see the Computer Science Internship Memorandum of Understanding (MOU), in PDF format or in MS-Word format.

Note 1: This is a course like any other where you need to be registered on time for the term (fall semester, spring semester, or summer) in which the internship will end. Many people have been denied registration for 391 because they

tried to register late, concentrating on the job, not the registration. Also you must get the approval of the Undergraduate Internship Coordinator first. Plan ahead.

Note 2: In order to get a grade recorded, the final reports of the student and the employer must be received by the Undergraduate Internship Coordinator before the last week of the finals week. Make sure the end date on the internship is early enough to fit with that. You may continue with more hours in your job after the for-credit internship part.

Note 3: You do not need to be classified as an "intern" in your job. It can be a regular job. The internship only comes from the relationship to your academic credit and the supporting documents indicated in the MOU. The overhead for the employer is comparable to a letter of recommendation.

Internship opportunities that the Computer Science Department has received are posted here and in the department on the fourth floor of Lewis Towers. Another resource is the university career and internship center. Students may also find a position through their own contacts.

3.70.3 Outcome

Application of classroom skills to real-world situations.

3.71 COMP 397: Research Seminar

Supplements CS Seminar by targeting students directly engaging in research and facilitating contributions in ongoing projects.

3.71.1 Credit Hours

1 (may be taken up to three times)

3.71.2 Description

Supplements CS Seminar by targeting students directly engaging in research and facilitating contributions in ongoing projects. Progress in outside projects tracked through milestones such as abstracts, small fellowship-style proposals, informal updates, and outcome-oriented goal setting. Emphasizes creating lasting impacts through establishing project continuity and presenting posters, papers, and slide shows.

3.71.3 Outcome

Regular progress on research projects and final presentations of results for demonstration to department faculty and students, and potential use at professional conferences.

3.71.4 Syllabi

None

3.72 COMP 398: Independent Study

The student and a sponsoring faculty member will determine an advanced topic for the student to work on.

3.72.1 Credit Hours

Variable (presently 1-3)

3.72.2 Prerequisites

Undergraduate Program Director approval

3.72.3 Description

Independent study of selected topics in computer science under the supervision of a faculty member.

The Process

- Find a professor who is interested in supervising what you want to do, or who has a project in mind that you are interested in doing. Look to professors you have had, or look at the research page on the web describing ongoing projects, or just ask professors.

- Discuss details of what your work will entail, the number of units of credit, and the expectations for grading.

- Write an email back to the professor, confirming the details of your discussion. Also include the semester you want to register, the number of units, and your student number (to aid the final registration). All this data is essential.

- The professor should respond, copying your data, and agreeing with your request, and the email should be copied to upd@cs.luc.edu.

- The department administrator should email all concerned when the registration is completed.

3.72.4 Outcome

Knowledge of an advanced topic.

3.73 COMP 399: Computer Science Seminar

3.73.1 Credit Hours

1 (may be taken up to three times)

3.73.2 Description

The department seminar is designed to bring together students interested in understanding and potentially engaging in extracurricular applications of computer science, with a focus on ongoing research. Speakers will include department faculty, research students, and invited outside speakers. Students will be asked to read relevant literature and participate in discussions.

3.73.3 Outcome

Knowledge of an advanced topic.

Students will be exposed to a wide range of topics in computer science, participate in discussions, and provide feedback to assess their general understanding of the presented material.

3.73.4 Syllabi

None

GRADUATE DEGREE PROGRAMS

4.1 Masters Programs Overview

The *M.S. in Information Technology* is management and business process facing. It is distinguished from a pure business degree by its requirement and frequent use of basic programming and other technical skills. If you end up managing technical workers, you will have an edge over those with pure business degrees. Read to see the general program and the special opportunities for a totally *online* version.

The *M.S. in Software Engineering* is focused on knowledge and abilities to produce high quality software for many different domains.

The *M.S. in Computer Science* is more foundational, requiring you to get experience with a broad range of core computer science concepts. It is our only program with a *Thesis Option*. Students from this program are the ones most likely to go on to a PhD program.

Masters Programs Prerequisites are required for all programs.

Undergraduates presently in our department with strong background are urged to consider overlapping one of these programs with their undergraduate work via the *BS/MS Dual Degree Programs*.

Electives: Computer Science is a broad and moving target. The full range of electives common to all programs is very helpful in providing breadth and flexibility. Only 3-5 courses out of 10 are specified or limited by the individual program. The overlap also means that it is generally fairly easy to switch programs if your interest is particularly stimulated by an elective more strongly associated with another program.

The *M.A. in Digital Humanities* is a separate interdisciplinary program, partnered with Computer Science.

An important separate document is the Graduate Handbook. See the description below.

4.1.1 Graduate Handbook

Tne Graduate Handbook is a separate document which has administrative and admissions information for the graduate program (also available in a pdf version).

Applicants should particular;ly note the Graduate Handbook sections for

- Prospective students

- Teaching assistantships and financial aid

Once you are enrolled, the section for ongoing reference will be **Regulations and Procedures**, and the topics are all listed in the table of contents at the beginning of the Graduate Handbook.

4.1.2 Time Requirements

Normally it takes 16 months for full-time students to earn a M.S. in Computer Science, Software Engineering, or Information Technology from Loyola. The M.S. in Computer Science may need extra time **if** the *Thesis Option* is chosen. Students who attend part time normally take 2-3 years. Though rarely an issue, there is a five year overall time limit, and this period may be extended only by special action of the Dean of the Graduate School.

4.1.3 Pursuing Multiple M.S. Degrees

We are often asked about whether it is possible to complete more than one M.S. degree (e.g. and M.S. in Computer Science and an M.S. in Software Engineering). The answer is no, unless you complete 54 hours of study (30 hours of courses for the each degree, except for 6 credits transferred into the second degree). It is impractical and expensive, and we don't think it serves students well. Our goal is to prepare you for a serious career (professional and/or academic) after completing *one* graduate degree.

4.1.4 Non-Degree Students and Continuing Education

Students who wish to continue taking courses may do so after completing their degree as non-degree students. We also encourage our students to maintain connections through our professional and social networking groups. Many of our students continue to be involved in research laboratories such as the Emerging Technologies Laboratory (see http://www.etl.luc.edu).

4.2 Masters Programs Prerequisites

All students need the equivalent knowledge of *COMP 170: Introduction to Object-Oriented Programming* and *COMP 271: Data Structures*. The topics for these courses are outlined in *Two Basic Programming Prerequisites*. This knowledge is prerequisite to essentially all the graduate courses in the Computer Science Department, with the exception of *COMP 417: Social, Legal, and Ethical Issues in Computing*.

Students in the Computer Science Graduate Program need the strongest preparation. Most will have an undergraduate degree or at least the first two years of foundational undergraduate courses. This would include the addition of equivalents of our *COMP 163: Discrete Structures* and *COMP 264: Introduction to Computer Systems* and *COMP 363: Design and Analysis of Computer Algorithms* to the two courses already listed above.

Software Engineering students are in the middle. The background for the Computer Science student is helpful, but adept transitioning students may manage with the two basic courses that the IT students need.

Many of our best students in Information Technology and Software Engineering have transitioned from other fields.

You can handle missing prerequisite knowledge in a number of ways:

1. Take them somewhere else (possibly cheaper and more convenient to your location, but possibly not as good or not covering mostly what we want)

2. Take them at Loyola: Take just the courses we want, with support of our faculty, giving you an opportunity to get into our community and build relationships with faculty and students.

   - This can be done before entering the Graduate program as a non-degree undergrad (cheaper undergraduate tuition).

   - You can get both prerequisites in one semester with our special compressed sections (section number 400) of *COMP 170: Introduction to Object-Oriented Programming* and *COMP 271: Data Structures*. See a fuller discussion below.

- The compressed prerequisites can be taken after admission to the graduate program. Then you become a full-time graduate student if you add a third course - a graduate course not requiring the prerequisites. (faster full-time finish!)

Compressed prep courses: In fall or spring you have the opportunity to take a compressed intensive sequence of the *COMP 170: Introduction to Object-Oriented Programming* and *COMP 271: Data Structures* prerequisites in one semester: two 7-8 weeks courses in succession, each with the same total contact hours of regular 15 week courses. To be a full-time graduate student at the same time, the most common additional course is the required *COMP 417: Social, Legal, and Ethical Issues in Computing*, which has no prerequisite. Note that 6 1/4 hours of class a week in the compressed prerequisites with all the work that goes with them does require some dedication! Many transitioning students with full-time jobs have found this to be *too much* to take on even without an additional grad course. You can see the speed of topics in the first course in fall 2016 by looking at the course website, http://anh.cs.luc.edu/388Intro.

Comp 170 does not require any background in programming, but if you have no background in programming, then it IS important to have a math background in which you are comfortable with the material in a precalculus course like our Math 118, which includes *significant manipulation of functions*. If you have a previous modest introduction to programming, including defining and using functions, then the formal math background is not so important.

Students who neither have this math background nor any programming at all may well want to get a more gentle introduction to programming first, as in our *COMP 150: Introduction to Computing*. It is offered fall and spring and online in the summer. If you do need the extra preparation like in Comp 150, then this will slow you down by a semester.

Please contact the Graduate Program Director with any questions at gpd@cs.luc.edu.

The next section, *Two Basic Programming Prerequisites*, lists more details of material you should know before taking graduate courses in our department.

4.2.1 Two Basic Programming Prerequisites

Knowledge of the content below is prerequisite for almost all of the graduate courses in the Computer Science Department.

The descriptions below are for the courses at Loyola, in a pretty standard introductory sequence. If you took these subjects elsewhere, you are not likely to have the exact same collection of topics, but if you are close, there should not be much problem filling holes as needed.

Expectations from Introduction to Computer Science, COMP 170

List of top ten concepts you should understand at the end of *COMP 170: Introduction to Object-Oriented Programming*, Introductory Object Oriented Programming (in Java, C#, C++ or a similar language):

1. Be able to distinguish between class and instance

    - Class is abstract representation or model (class aka type, abstract type, factory for objects). Class has field definitions (or attribute definitions) and operations (or methods) to implement its behaviors.

    - Instance (aka class instance, variable) is an actual, specific, concrete thing and has specific field values (or attribute values).

    - All instances of a class share the same operations and attribute definitions.

2. Know syntax rules and coding style

    - Learn the basic picky rules of syntax and use them correctly. Key items: placement of ";" use of case ("ClassSchedule" vs. "classSchedule"); distinguish declarations and executable statements.

    - Format code readably according to agreed upon style.

3. Use conditional control structures fully and correctly

- Construct programs with "if….then", and "if….then…else". Understand nesting of control statements and blocks ("{" and "}").

- Briefly, "Nesting, blocks, no dangles".

4. Construct loops with control structures

- Use "while", "do", and "for" control structures to implement algorithms successfully.

- Briefly, "avoid never ending loops, avoid off-by-one problems"

5. Use the primitive types and operators appropriately

- Understand the use and limitations of the primitive types.

- Know the most useful arithmetic, relational, and logical operators and use them in expressions, with parentheses when necessary.

6. Lean how to create good OO encapsulation

- Achieve a deep understanding of the proper use of the access modifiers "private", and "public".

- Be able to design well structured classes.

- Know when and how to use "static".

7. Construct and understand methods fully

- Define and use methods, including formal parameters and return values.

- Distinguish between formal parameters (aka parameters) and arguments in the call to a method.

- Comprehend scope and lifetime of objects. Understand visibility of objects and how to implement finding the right object at the necessary time.

- Know how to ask a particular instance of a class to do something. Know how to construct associations between classes.

- Understand what is happening is a nested stack of method calls.

8. Be knowledgeable about important library container classes and arrays

- Be able to declare and use container objects like for lists and arrays. Understand how to find things in containers and arrays. Use index variables effectively.

9. Perform simple input/output processing

- Lean how to do basic input and output statements with an interactive user and with files. Be able to work with Strings.

10. Understand OO Interfaces

- Know how to define and use simple interfaces (example - List in Java).

Expectations from Data Structures, COMP 271

1. OOP

- Be able to write interface/abstract classes and implementations of data structures.

- Understand subclassing mechanisms.

- Understand overloading, overriding, dynamic binding, boxing and unboxing in OOP languages.

2. Mathematical context

- Big-Oh notation, worst-case run time complexity and storage complexity of algorithms.

- Be able to do performance analysis of code using big-Oh.

3. Data structures

   - Know and be able to use and implement abstract data types such as stacks, arraylists, vectors, queues, linked lists, hash tables, and hash sets.

   - Understand the time complexities of operations such as insert, delete, add, retrieve, on the above data structures and be able to obtain time complexities of other operations on similar, simple data structures.

   - Knowledge of hash functions, trees, graphs, and two dimensional arrays.

   - Knowledge of algorithms for problems including searching, and sorting (Quicksort, Mergesort, Heapsort etc.), string manipulations.

4. Core programming language concepts

   - Knowledge of programming language storage paradigms such as static, stack, and heap.

   - Be able to understand and use recursion in various problems.

4.3 M.S. in Computer Science

4.3.1 Overview

The M.S. in Computer Science provides foundations of Computer Science, while acknowledging that Computer Science is key to algorithms in the sciences. Other programs in the Computer Science Department with different foci are *M.S. in Information Technology* and the *M.S. in Software Engineering*.

4.3.2 Curriculum

This program is for students entering the program in Spring 2014 or later. If you were enrolled in the program prior to Spring 2014 you may use the older degree requirements located here.

Features Common to all Programs

- See the *Masters Programs Prerequisites*.

- Each program requires a total of 30 credit hours, generally 10 three-credit courses.

- Students can receive up to 6 credit hours (replacing two of our courses) for graduate work taken elsewhere: See the transfer credit section of the Graduate Student Handbook for details.

- **Students who have taken undergraduate Loyola classes**: Many 400-level courses in the department have closely related 300-level analogues, e.g. *COMP 443: Computer Networks* and *COMP 343: Introduction to Computer Networks*. Students who enter the MS program after taking a Loyola course in this category must choose to take 400-level courses that are not closely related to any 300-level courses taken earlier, unless they have GPD permission.

Note that Computer Science has more prerequisites than the other programs.

Overall, to achieve depth and breadth, Computer Science students must complete the following:

- Three (3) required courses.

- Three (3) restricted electives.

- Four (4) courses or 12 credits that are general electives.

Major Requirements (9 credits)

The following three courses are required by the program:

- *COMP 417: Social, Legal, and Ethical Issues in Computing*
- *COMP 413: Intermediate Object-Oriented Development*
- *COMP 460: Algorithms and Complexity* (note important prereq *COMP 363: Design and Analysis of Computer Algorithms*)

The Graduate Program Director may approve the replacement of *COMP 413: Intermediate Object-Oriented Development* by an additional course on the Restricted Electives list in cases where the student's primary program focus is not on software development.

Restricted Electives (9 credits)

MS-CS students must also take three courses from the following list:

- *COMP 410: Operating Systems*
- *COMP 433: Web Services Programming*
- *COMP 436: Markup Languages*
- *COMP 439: Distributed Systems*
- *COMP 442: Server-Side Software Development*
- *COMP 443: Computer Networks*
- *COMP 453: Database Programming*
- *COMP 464: High-Performance Computing*
- *COMP 471: Programming Languages*
- *COMP 473: Object-Oriented Programming*
- *COMP 474: Software Engineering*

Note that several of these courses require *COMP 413: Intermediate Object-Oriented Development* as prerequisite.

General Electives (12 Credits)

The elective course options are common for all programs, differing only in the total number of credits required.

Electives can be any COMP 400 level class or any CSIS class (courses actually taught in the Business School).

Students may take up to a maximum of 6 credit hours of *COMP 490: Independent Project* and/or *COMP 499: Internship*.

There are numerous options for independent study, including a programming project, research, or a service-oriented project.

Thesis Option

MS students in the Computer Science degree program (only) may elect the MS thesis option.

Course work is strongly recommended over the thesis option, especially for those not planning on a research-oriented career. Many students pursuing the thesis option, therefore, would be considering a PhD program at another institution.

Students wishing to do a thesis should discuss this option as early as possible with the GPD.

Theses may involve research in purely theoretical computer science (for example, development or analysis of algorithms), or may involve development of a software package, or may involve instrumentation, measurement and analysis of existing systems (for example, studying network performance). Because of this wide range, there is no one formal course in research methods. Courses in the restricted-electives list above contain a significant component of area-specific integrated research-methods material. Students interested in writing a thesis are strongly urged to seek advising from the GPD or other faculty as early as possible as to which electives in this group will be the most appropriate for the student's proposed area of research.

Here is an outline of the steps toward your thesis. The ones in boldface are formal steps with documentation required by the Graduate School:

1. If you wish to write a thesis, first identify a faculty advisor and select a tentative topic or area of research. The existing program allows you to take up to 6.0 hours of *COMP 490: Independent Project*. You will typically begin their research program in such a course, though you may also identify an advisor and select a tentative topic as part of a conventional classroom course.

2. Thesis/Dissertation Committee Recommendation

   - The next step is for you to secure permission to pursue the thesis option from the Graduate Program Director.

   - The Graduate Program Director, in consultation with you and your chosen advisor, recommends a thesis committee to the Graduate School. The committee will consist of at least three faculty members; normally the committee director will be the advisor.

   - **For the formal Graduate School process you create the committee at the gsps link** https://gsps.luc.edu/. An email will then be sent to the director for approval and then the GPD.

   - You will receives an email if your committee is formally approved by the Graduate School.

   At least 50% of the committee must be comprised of Loyola graduate faculty; the director of the committee must have full graduate faculty status – see http://www.luc.edu/gradschool/about_facultystaff.shtml for the current list of full members.

3. You may finish while still taking *COMP 490: Independent Project* or other for-credit courses. Alternately you may still be continuing with your research in subsequent semesters, after finishing all your required for-credit courses. Then, assuming your *committee is approved*, you maintain full-time status by getting the GPD to register you for the zero-credit-hour Comp 595: Thesis Supervision, if available, or Comp 605: Masters Study, if Comp 595 is not listed. Students may register for any number of semesters of Comp 595/605, subject to time-to-degree-completion constraints.

4. Once you have your thesis committee approved, the thesis becomes a degree requirement. (This is important for international students.) You may, however, petition to revert back to non-thesis status; this requires permission of the Graduate Program Director. At that point, you would be able to graduate without writing a thesis, if the coursework requirements were met. No reimbursement or credit will be received for any Comp 595, Comp 605, or other thesis-specific courses taken.

5. Ballot for the Approval of a Thesis/Dissertation Proposal

   - You will then prepare a formal research proposal, in consultation with your advisor. This proposal must be submitted to your committee for review. This sequence is monitored through gsps.

   - If you are submitting your proposal to the Institutional Review Board (IRB), you must have approval or exemption *before* the Graduate School approves your proposal. You do not need to submit proof, Graduate School will confirm. (This step is required for the use of human and animal subjects, and is not common for Computer Science.)

   - **A simple one to two paragraph abstract must be included in the proposal form on gsps.**

- You will get notified once all members of the committee, the GPD, and the Graduate School approve the proposal through gsps.

- You are now ready to "conduct research" for the project.

6. Ballot for Text and Oral Defense Form

   - Upon completion of your thesis, you will be required to formally defend your research. Schedule this with your committee. You thesis should be in nearly final form.

   - Typically you should give the committee three weeks to read the final draft of the thesis before the defense date.

   - **For this requirement, you must download the defense ballot** http://www.luc.edu/media/lucedu/gradschool/pdfs/T%20&%20D%20defense%20ballot--DB.pdf and bring it with you to your defense. Your director and other committee member(s) will sign the ballot – this ballot then needs to go to the GPD for final approval.

   - The committee may require modifications before approving the thesis, or possibly reject it.

   - Once approved, the GPD will upload the ballot in gsps for Graduate School approval.

   - You will be notified once the process is complete.

7. Formatting the Thesis/Dissertation

   - Every thesis/dissertation needs to be formatted according to the rules stated in the Graduate School's formatting manual http://www.luc.edu/gradschool/formatting.shtml.

   - **Format check is a required step**; the deadlines, depending on the conferral date, are posted on the Key Dates and Deadlines page on the Graduate School website. These deadlines are well before the end of the semester - be sure to check and satisfy them.

   - **Final Copy, both electronic and hard copies also have deadlines**, depending on the conferral date.

4.4 M.S. in Information Technology

4.4.1 Overview

The Master of Science in Information Technology is designed for current and aspiring professionals in charge of developing, implementing, operating, and managing information systems in a variety of organizations. Students in this program will gain a broad technical understanding of current and emerging technologies in the industry, a familiarity with systems engineering concepts, and a solid foundation in net-centric computing. They will also have a firm grasp of current and future effects of the convergence of the telecommunications, media, and information technology sectors.

Courses may be taken in Chicago, online or at our *new Vernon Hills Campus* campus.

4.4.2 Entirely Online Option

The program can now be completed entirely online with some decrease in available electives. Networking and security tracks may be completed online and full-time. The management tracks can be completed online at least part-time over two years. Students taking mostly regular face-to-face courses are also welcomed into online courses when they find that convenient, but students signed up explicitly for the entirely online option *get the advantage of being able to be registered before other students* who might happen to choose to take some of their courses online. Student in the entirely online program should *contact the GPD with course choices* three days before the usual start of registration.

4.4.3 Vernon Hills Campus

The MS IT degree can be obtained not only at the Chicago campus and online, but also with courses in Vernon Hills, at the Cuneo Mansion. See the cuneo-schedule. students may also mix the location and form of their classes. Students taking courses entirely at Vernon Hills will need to be part-time over two years to complete the program.

4.4.4 Curriculum

Most students complete the program full-time, taking mostly face-to-face courses.

Features Common to all Programs

- See the *Masters Programs Prerequisites*.

- Each program requires a total of 30 credit hours, generally 10 three-credit courses.

- Students can receive up to 6 credit hours (replacing two of our courses) for graduate work taken elsewhere: See the transfer credit section of the Graduate Student Handbook for details.

- **Students who have taken undergraduate Loyola classes**: Many 400-level courses in the department have closely related 300-level analogues, e.g. *COMP 443: Computer Networks* and *COMP 343: Introduction to Computer Networks*. Students who enter the MS program after taking a Loyola course in this category must choose to take 400-level courses that are not closely related to any 300-level courses taken earlier, unless they have GPD permission.

Overall, to achieve depth and breadth, Information Technology students must complete the following:

- One (1) required course addressing ethical and social issues in the computing field.

- Three (3) courses within *one* of the following tracks: Enterprise Information Management, Technology Management, Information Assurance, or Enterprise Networking.

- Six (6) courses or 18 credits that do not need to be in any track. These electives can be any 400-level course offered by the department, with approval by your advisor.

The courses listed below can be counted toward the different tracks. Any other COMP course (400 level or higher) can count as an elective towards the required 30 credits. Special permission is required to count any other course as a track requirement or as an elective. **CSIS courses are no longer being offered. The corresponding INFS courses may be taken for MSIT credit with permission only. We will not withdraw acceptance of courses allowed before that you already took.**

Note that you only need three courses in *some* track. There is no need to satisfy more than one track.

Major Requirements (3 credits)

- *COMP 417: Social, Legal, and Ethical Issues in Computing*

Tracks (9 credits)

Students starting before Spring 2017 may also include courses from the old EIM or TM tracks (*M.S. in Information Technology: Old EIM and TM Tracks*). Anyone may follow the tracks listed below:

- Three (3) courses from any *one* of the following Tracks must be taken.

Enterprise Information Management (EIM) Track

- *COMP 453: Database Programming*
- COMP 488-301/302: Data Mining

- COMP 488-305: Data Administration
- COMP 488-325: Rapid Application Development(.Net)
- COMP 488-379: Machine Learning

Technology Management (TM) Track

- *COMP 420: Software Systems Analysis*
- *COMP 477: IT Project Management*
- COMP 488-472: Organizational Change and Development
- COMP 488-480: Operations Management

Information Assurance (IA) Track

- *COMP 431: Cryptography*
- *COMP 443: Computer Networks*
- *COMP 447: Intrusion Detection*
- *COMP 448: Network Security*
- *COMP 449: Wireless Networks and Security*

Enterprise Networking (EN) Track

- *COMP 443: Computer Networks*
- *COMP 446: Telecommunications*
- *COMP 448: Network Security*
- *COMP 449: Wireless Networks and Security*
- *COMP 451: Network Management*

General Electives (18 Credits)

The elective course options are common for all programs, differing only in the total number of credits required.

Electives can be any COMP 400 level class or any CSIS class (courses actually taught in the Business School).

Students may take up to a maximum of 6 credit hours of *COMP 490: Independent Project* and/or *COMP 499: Internship*.

There are numerous options for independent study, including a programming project, research, or a service-oriented project.

4.5 M.S. in Information Technology: Old EIM and TM Tracks

The following old EIM and TM tracks include courses offered by the Quinlan School of Business.

Enterprise Information Management (EIM) Track

- *COMP 453: Database Programming*
- csis494
- csis796

Technology Management (TM) Track

- *COMP 477: IT Project Management* / csis484

- csis483

- csis794

- csis472

4.6 M.S. in Software Engineering

4.6.1 Overview

The M.S. in Software Engineering caters to students interested in software engineering and also other contemporary topics of long-term value to the industry. Most early careers in the industry are based on either software development or managerial aspects of software development. This degree places more weight on software development. For those seeking a greater focus on managerial aspects, we offer the *M.S. in Information Technology*.

4.6.2 Curriculum

This program is for students entering the program in Spring 2014 or later. If you were enrolled in the program prior to Spring 2014 you may use the older degree requirements located at here.

Features Common to all Programs

- See the *Masters Programs Prerequisites*.

- Each program requires a total of 30 credit hours, generally 10 three-credit courses.

- Students can receive up to 6 credit hours (replacing two of our courses) for graduate work taken elsewhere: See the transfer credit section of the Graduate Student Handbook for details.

- **Students who have taken undergraduate Loyola classes**: Many 400-level courses in the department have closely related 300-level analogues, e.g. *COMP 443: Computer Networks* and *COMP 343: Introduction to Computer Networks*. Students who enter the MS program after taking a Loyola course in this category must choose to take 400-level courses that are not closely related to any 300-level courses taken earlier, unless they have GPD permission.

Overall, to achieve depth and breadth, Software Engineering students must complete the following:

- Two (2) required courses addressing ethical and social issues in the computing field and a firm base in object oriented programming.

- Three (3) courses from restricted categories to ensure a strong software engineering center, while still allowing a considerable flexibility of interests.

- Five (5) courses or 15 credits that are general electives.

Major Requirements (6 credits)

- *COMP 413: Intermediate Object-Oriented Development* (prereq to some other courses - take early!)

- *COMP 417: Social, Legal, and Ethical Issues in Computing*

Restricted Electives (9 credits)

MSSE students must take at least one course from the following Group 1 list:

Group 1

- *COMP 424: Client-Side Web Design*
- *COMP 433: Web Services Programming*
- *COMP 434: Enterprise Software Development*
- *COMP 437: Concurrent Programming*
- *COMP 439: Distributed Systems*
- *COMP 442: Server-Side Software Development*
- *COMP 460: Algorithms and Complexity* (note special prereq, *COMP 363: Design and Analysis of Computer Algorithms*)
- *COMP 471: Programming Languages*
- *COMP 473: Object-Oriented Programming*
- *COMP 474: Software Engineering*

Most of these Group 1 courses require *COMP 413: Intermediate Object-Oriented Development* as a prerequisite. Excepetions are Comp 424, 474 and 460.

MSSE students must also take two additional courses from either the list above or the following Group 2 list:

Group 2

- *COMP 410: Operating Systems*
- *COMP 412: Free/Open Source Computing*
- *COMP 420: Software Systems Analysis*
- *COMP 436: Markup Languages*
- *COMP 441: Human-Computer Interface Design*
- *COMP 453: Database Programming*
- *COMP 464: High-Performance Computing*

If a student enters the program with an academic record of success in a course similar to *COMP 413: Intermediate Object-Oriented Development*, or if the student can demonstrate programming experience with the concepts of *COMP 413: Intermediate Object-Oriented Development*, then the Graduate Program Director may substitute for the *COMP 413: Intermediate Object-Oriented Development* requirement a course in Group 1 or 2 that has *COMP 413: Intermediate Object-Oriented Development* as a prerequisite.

The Computer Science department may declare that other courses (eg new courses and individual *COMP 488: Topics in Computer Science* offerings) may count as members of either Group 1 or Group 2.

General Electives (15 Credits)

The elective course options are common for all programs, differing only in the total number of credits required.

Electives can be any COMP 400 level class or any CSIS class (courses actually taught in the Business School).

Students may take up to a maximum of 6 credit hours of *COMP 490: Independent Project* and/or *COMP 499: Internship*.

There are numerous options for independent study, including a programming project, research, or a service-oriented project.

4.7 M.A. in Digital Humanities

The Computer Science department is a partner in the MA in Digital Humanities program. While not a department-specific program, we encourage students with interdisciplinary interests at the intersection of culture and technology to consider this degree, which allows one to elect courses from the CS department.

The Digital Humanities M.A. program at Loyola University's Center for Textual Studies and Digital Humanities (CTSDH) combines theoretical and practical courses. Its aims are ultimately practical and professional, training new digital specialists for the growing knowledge and information economy and today's research in humanities disciplines. Because the nature of much Digital Humanities work is constructive and project-based, students in the M.A. program will be given hands-on training in workshop or seminar-based classes, training in text editing and text encoding, electronic publishing and platforms, programming, interface design, project management, and archive construction. At every stage, team-based collaborative learning will be encouraged, in class projects, for example, and potentially in the required electronic thesis project. But the MA program also explores theoretical, critical, social, and ethical contexts for thinking about Digital Humanities research and applications, including issues of intellectual property, data and privacy, public access and preservation.

For details about this program, see http://www.luc.edu/ctsdh/academics/maindigitalhumanities/.

FIVE

GRADUATE COURSES

For administrative information on course loads, registration, and course selection, please see the Regulations Section of the Graduate Student Handbook.

5.1 COMP 409: Advanced Numerical Analysis

5.1.1 Aliases

MATH 409: Advanced Numerical Anaylsis

5.1.2 Credit Hours

3

5.1.3 Prerequisites

MATH 212: Linear Algebra

MATH 264: Ordinary Differential Equations

COMP 271: Data Structures

5.1.4 Course Description

Introduction to error analysis, numerical solution of equations, interpolation and approximation, numerical differentiation and integration, matrices and solution of systems of equations, numerical solution of ordinary and partial differential equations.

5.1.5 Syllabi

No recent syllabi available.

5.2 COMP 410: Operating Systems

5.2.1 Credit Hours

3

5.2.2 Prerequisites

COMP 374: Introduction to Operating Systems

5.2.3 Description

This is a course in Operating Systems discussing both standalone and distributed environments. The focus of the course is to understand the underlying technologies that make contemporary operating systems work efficiently. We will discuss processes, threads, synchronization, I/O, file systems, memory management, transactions and system coordination techniques. Through this course we will discover how these technologies are integrated into the systems we use today and then utilize these technologies and apply them to practical applications.

This is NOT a programming intensive course, however, students will be expected to complete some programming in C with plenty of examples and assistance along the way. You certainly don't need to know how to program in C today. In addition, the completion of a technical paper on an OS related subject will also be expected.

Everyone currently in, or planning to enter the IT field should have a grasp on these components as they effect every area of the day to day operation of IT technology. Reference systems will include both Linux and Windows.

Since this course is an advanced study of Operating Systems, it is assumed that the student has already taken an undergraduate Operating System course. The course covers:

processes and threads (particularly the concepts of kernel threads, user threads, and lightweight processes); process and thread scheduling (time- sharing, real-time); file systems (including network file systems); communications (including RPC and sockets); synchronization (introducing mutexes and condition variables in addition to semaphores); Inter Process Communications (POSIX pipes, FIFOs, message queues, shared memory); security issues (including authentication and access control).

5.2.4 Syllabi

Semester/Year	Instructor	URL
Spring 2013	Joe Kaylor	https://luc.box.com/s/7krnwnsup8j13ekyeelsju7p0mt902wi
Spring 2012	Joe Kaylor and George K. Thiruvathukal	https://luc.box.com/s/7krnwnsup8j13ekyeelsju7p0mt902wi

5.3 COMP 412: Free/Open Source Computing

5.3.1 Credit Hours

3

5.3.2 Prerequisites

COMP 271: Data Structures

5.3.3 Description

This course will cover the fundamentals of Free and Open Source software development. Topics to be addressed include licensing, Linux, typical software development tools (e.g. compilers, scripting languages, build tools, and version control software), applications, and techniques for managing remote servers. Students will work on a significant developmentproject involving free and open-source software and learn how to participate in open-source projects effectively.

5.3.4 Syllabi

Semester/Year	Instructor	URL
Summer 2014	George K. Thiruvathukal	https://luc.box.com/s/w7o4ewcry5ogfiruuhulrott87rtwt8s
Spring 2014	Matt Butcher	https://luc.box.com/s/w7o4ewcry5ogfiruuhulrott87rtwt8s

5.4 COMP 413: Intermediate Object-Oriented Development

5.4.1 Credit Hours

3

5.4.2 Prerequisites

COMP 271: Data Structures (strictly enforced)

5.4.3 Description

Object-orientation continues to be a dominant approach to software development. This intermediate programming-intensive course studies the use of classes and objects with an emphasis on collaboration among objects.

Overall Series of Object-Oriented Courses

- *COMP 170: Introduction to Object-Oriented Programming* (CS1) - simple objects representing scalars

- *COMP 271: Data Structures* (CS2) - collections of simple objects

- *COMP 313: Intermediate Object-Oriented Development* / *COMP 413: Intermediate Object-Oriented Development* - complex, interacting objects; basic design patterns

- *COMP 373: Objects, Frameworks, and Patterns* / *COMP 473: Object-Oriented Programming* - advanced design patterns and topics such as AOP(Aspect-Oriented programming)

COMP 313: Intermediate Object-Oriented Development / *COMP 413: Intermediate Object-Oriented Development* is also a prerequisite for other advanced software courses. Students interested in advanced software courses are encouraged to take *COMP 313: Intermediate Object-Oriented Development* / *COMP 413: Intermediate Object-Oriented Development* as soon as they have completed *COMP 271: Data Structures* so as to be eligible for these further courses.

Course Topics

- Data Structures of various types – linear vs. nonlinear, indexing vs. non-indexing, position vs. value-oriented

- Advanced Java, e.g. interfaces, annotations, exceptions, generics, collections, boxing/unboxing, array objects

- Object Modeling – UML, use cases and activity diagrams, class diagrams, archetypes, interaction diagrams

- Design by contract, interfaces, refactoring & generalization, design patterns (Adapter, Decorator, Composite, Strategy, Iterator, Abstract Factory, Visitor, . . .)

- Agile Development Process – evolutionary design, test-driven development, refactoring, . . .

- Tools – Eclipse, Subversion, JUnit, JMock, Ant, . . .

- Techniques – object pooling, garbage collection, performance profiling (NetBeans)

5.4.4 Outcome

A thorough understanding of the principles of object-orientation: abstraction, delegation, inheritance, and polymorphism; exposure to basic design patterns; programming experience in mainstream object-oriented languages such as C++ and Java.

You will take your software development abilities to the next level by building on your knowledge of data structures.

You will learn to design and implement more complex programs using good software engineering practices, including:

- Designing with interfaces and composition

- Design patterns

- Refactoring

- Test-driven development (TDD)

5.4.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Yacobellis	https://luc.box.com/s/m24hdedq5ghw8ecgr7gjtdtkg71juvhy
Fall 2013	Dr. Läufer	https://luc.box.com/s/m24hdedq5ghw8ecgr7gjtdtkg71juvhy

5.5 COMP 417: Social, Legal, and Ethical Issues in Computing

5.5.1 Credit Hours

3

5.5.2 Prerequisites

None

5.5.3 Description

This course will explore a variety of ethical and legal issues facing those who use or program computers. Issues can be divided broadly into professional ethics, dealing with the ethical responsibilities of the programmer, and social issues, dealing with concerns we all have as citizens.

5.5.4 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Dordal	https://luc.box.com/s/l777d6kwy0kw6ek2fcw302frnwhbuhhn

5.6 COMP 418: Combinatorial Mathematics

5.6.1 Credit Hours

3

5.6.2 Prerequisites

MATH 162: Calculus II

5.6.3 Description

Combinatorics is a branch of mathematics with broad areas of application. There are important uses of combinatorics in computer science, operations research, probability, and statistics. Theoretical thermodynamics uses combinatorial theory to describe ideas such as entropy. Combinatorial analysis is a cornerstone of the study of error-correcting codes; these codes are used to transmit information from deep space or to protect the quality of music on compact discs. Our course will mainly focus on describing and/or counting complicated sets. Often questions which begin "How many ways can you...?" or "How many steps does it take to...?" are answered using combinatorial analysis. Such questions on the surface may appear rather uninteresting, but one can quickly get to questions that are quite engaging. What gambler wouldn't want to understand the odds at winning a poker hand?

We plan to examine the following topics: permutations and combinations, the inclusion-exclusion principle and other general counting techniques, partitions, generating functions, recurrence relations, Burnside's Theorem, the cycle index, and Polya's formula. Other topics may be included as time permits. Emphasis will be on examples rather than theory.

This course is a combined undergraduate/graduate course. The requirements of the course for the graduate students will be different from the requirements for the undergraduates. The material should be comprehensible for any student who has completed MATH 162.

5.6.4 Other Information

Text: Applied Combinatorics, 4th edition, by Alan Tucker.

5.6.5 Syllabi

No recent syllabi available.

5.7 COMP 420: Software Systems Analysis

5.7.1 Credit Hours

3

5.7.2 Prerequisites

COMP 271: Data Structures

5.7.3 Description

This course uses Unified Modeling Language notation to model the early software analysis and design phase. Object technology is critical to the understanding of the process of capturing business requirements and the course uses commercial software tools to perform systems analysis and design.

5.7.4 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Conrad Weisert	https://luc.box.com/s/tlysz13he39x71rw7dezny3mljzwx1xl

5.8 COMP 421: Mathematical Modeling and Simulation

5.8.1 Credit Hours

3

5.8.2 Prerequisites

MATH 215: Object Oriented Math Programming or *COMP 215: Object-Oriented Programming with Mathematics* or *COMP 170: Introduction to Object-Oriented Programming*

MATH 132: Applied Calculus II or MATH 162: Calculus II

STAT 203: Statistics

5.8.3 Description

The tool of simulation is probably the most useful tool for analyzing problems which arise in real applications which are much too complicated to be analyzed mathematically. One example is the study of inventory control policies. When should the inventory be checked? How many units should be ordered? What are the anticipated consequences of these decisions? We simulate the random character of the demand for the product and the delay in receiving orders of the product. The objective is to provide management with relevant information for constructing an acceptable inventory policy.

5.8.4 Syllabi

No recent syllabi available.

5.9 COMP 422: Software Development for Wireless/Mobile Devices

5.9.1 Credit Hours

3

5.9.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

5.9.3 Description

This course will focus on the unique challenges, methods, tools, and technologies for developing software applications for wireless and mobile devices, such as personal digital assistants (PDA) and smart mobile phones. Topics include user interface design for smallscreen, multi-channel devices, programming techniques and memory management for devices with limited memory and processing power, data synchronization for mobile databases, and wireless network programming.

5.9.4 Syllabi

Semester/Year	Instructor	URL
Spring 2009	Dr. Honig	https://luc.box.com/s/sm75okfbnv8b7fn0z55cw4rkjozju31n

5.10 COMP 424: Client-Side Web Design

5.10.1 Credit Hours

3

5.10.2 Prerequisites

COMP 271: Data Structures

5.10.3 Description

This course provides an in-depth study of the concepts and methods required for the design and implementation of the presentation layer of a web application. Topics include visual design, usability design, multi-channel and multi-modal applications, markup of static and dynamic content, content transformation, client-side executable content including client-side scripting and embedded applets, and web-based content management systems. Coursework includes several substantial programming projects (using technologies such as XHTML, XSTL, DHTML, JavaScript, PHP/ASP/JSP, Flash, and Zope/Plone).

5.10.4 Syllabi

Topic	Semester/Year	Instructor	URL
Client Side Web Design	Spring 2016	Nicholas Hayward	https://luc.box.com/s/ eslnd9ypeus1j8xpzqci4ujuhpey6boq

5.11 COMP 428: Algebraic Coding Theory

5.11.1 Aliases

MATH 428: Algebraic Coding Theory

5.11.2 Credit Hours

3

5.11.3 Prerequisites

MATH 212: Linear Algebra

COMP 163: Discrete Structures or MATH 313: Abstract Algebra

5.11.4 Description

In this course, major types of error-correcting codes, encoding and decoding, and their main properties will be studied. The codes examined will include the Hamming, Golay, BCH, cyclic, quadratic residue, Reed-Solomon, and Reed-Muller codes.

5.11.5 Outcome

Students will learn both the theory and application of error-correcting codes.

5.11.6 Syllabi

No recent syllabi available.

5.12 COMP 431: Cryptography

5.12.1 Prerequisites

Mathematics (one of *COMP 163: Discrete Structures*, MATH 313: Abstract Algebra or MATH 201: Elementary Number Theory)

Programming *COMP 271: Data Structures*

5.12.2 Description

This course introduces the formal foundations of cryptography and also investigates some well-known standards and protocols. The intended audience is senior undergraduate and beginning graduate students. The course will include topics selected from the following: information-theoretic security, private key encryption, DES, public key encryption, background on modular arithmetic, RSA, hashing and message authentication codes (MACs), digital signatures, DSS, key distribution and management, PGP, network security, and Fiat-Shamir protocol.

5.12.3 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Stephen Doty	https://luc.box.com/s/i5gsso8w12vwiog80u1aijcz2e6h07kf

5.13 COMP 433: Web Services Programming

5.13.1 Credit Hours

3

5.13.2 Prerequisites

- *COMP 413: Intermediate Object-Oriented Development*
- *COMP 442: Server-Side Software Development* (highly recommended)
- instructor permission if prerequisite is missing

All MS students are expected to have completed the undergraduate prerequisites:

- *COMP 388: Foundations of Computer Science 1* is an accelerated version of *COMP 163: Discrete Structures* and *COMP 170: Introduction to Object-Oriented Programming*
- *COMP 388: Foundations of Computer Science 2* is an accelerated version of *COMP 264: Introduction to Computer Systems* and *COMP 271: Data Structures*

Please note that MS IT students are expected to complete *all* prerequisites before taking any course in CS or the Quinlan School of Business. (This includes any additional prerequisites required by Quinlan.)

5.13.3 Description

Web services are building blocks for enterprise applications that use open data exchange standards and transport protocols to exchange data with calling clients. This course studies the architectures, frameworks, and tools required to develop and compose web services and clients, as well as integrate service-oriented systems with legacy systems.

5.13.4 Syllabus

Offering	Instructor	URL
Fall 2015	Berhane Zewdie	https://luc.box.com/s/zy8564ok8cmwiu1tk9xwade93dtu0pss

5.14 COMP 434: Enterprise Software Development

5.14.1 Credit Hours

3

5.14.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

5.14.3 Description

The course shows how to use Enterprise JavaBeans to develop scalable, portable business systems. The technologies taught in the course include: component models, distributed objects, asynchronous messaging, and component transaction monitors. The course covers architecture of EJB, entity and message and session beans.

5.14.4 Syllabi

No recent syllabi available.

5.15 COMP 436: Markup Languages

5.15.1 Credit Hours

3

5.15.2 Prerequisites

COMP 271: Data Structures

5.15.3 Description

This course covers Extensible Markup Language (XML) and its applications. This course will cover the core XML component frameworks, including XSLT (a transformational approach) and the various W3C specifications for manipulating XML documents programmatically, including the DOM and SAX frameworks. As well, this course will cover some advanced topics, including how to manage large XML documents and integration with databases. Please note that HTML will be occasionally used in this course; however, this course is not about HTML and students may want to acquire an HTML book and study it briefly before taking this course. This course is not about making cool web pages. The course has almost nothing to do with web pages and is focused more on modeling and the emergent notion of web services.

Students should expect the programming to be somewhat involved (intermediate to advanced). Most of the programming is based on straightforward data structures, such as trees, lists, and maps (collections found in Java). You are encouraged to study these topics on your own. Little time will be spent rehashing this preliminary knowledge.

5.15.4 Syllabus

Offering	Instructor	URL
Fall 2013	Matt Bone	https://luc.box.com/s/4urhyalrnsvilautu8tgwb3vsl5hv480
Spring 2012	Dr. Thiruvathukal	http://markuplanguages.cs.courseclouds.com

5.16 COMP 437: Concurrent Programming

5.16.1 Credit Hours

3

5.16.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development

5.16.3 Description

This course studies the architecture, design, and implementation of concurrent software systems. Process algebras, formal specification, and testing are used as tools in the engineering of concurrent systems; event-based programming frameworks and thread libraries are employed in the implementation of such systems. Coursework includes several substantial programming projects (in a language such as Java) involving applications of concurrency and event-driven programming such as graphical user interfaces and distributed services using Remote Method Invocation (RMI).

5.16.4 Syllabi

No recent syllabi available.

5.17 COMP 439: Distributed Systems

5.17.1 Credit Hours

3

5.17.2 Prerequisites

COMP 313: Intermediate Object-Oriented Development or *COMP 374: Introduction to Operating Systems*

5.17.3 Description

This course is designed as a modern discussion of distributed computing systems, which represent one of the most important areas in academic and business computing today. Topics covered include distributed computing, interactive services, collaborative computing, and peer-to-peer sharing.

Various distributed frameworks and technologies will be explored, e.g. DNS, CORBA, Java RMI, SOAP (XML-based RPCs), and Globus.

5.17.4 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Dr. Thiruvathukal	https://luc.box.com/s/s42vfem0702wvlo0fje3j8kff2hdjvmp

5.18 COMP 441: Human-Computer Interface Design

5.18.1 Credit Hours

3

5.18.2 Prerequisites

COMP 271: Data Structures

5.18.3 Description

This course studies the interaction between humans and computer-based systems. The course will provide students with the methods for evaluating, designing, and developing better interfaces between humans and systems. Students will acquire an awareness of different design and evaluation methods as well as practical, effective, and cost-conscience methods for improving systems and their interfaces.

5.18.4 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Nicholas Hayward	https://luc.box.com/s/x6ygjsx6f8nxqj4gjxl8xwuk3grpgkf2

5.19 COMP 442: Server-Side Software Development

5.19.1 Credit Hours

3

5.19.2 Prerequisites

COMP 264: Introduction to Computer Systems and *COMP 313: Intermediate Object-Oriented Development*

5.19.3 Description

This course studies the architecture, design, and implementation of multitiered server-based software systems. Each tier is studied along with the pertinent mechanisms: markup languages in the client tier, web application frameworks in the web tier, and server-side components in the business logic and persistence tiers. Pervasive issues such as integration, testing, security, and performance are discussed. Coursework includes several substantial programming projects (using a platform such as Java 2 Enterprise Edition).

5.19.4 Syllabi

Semester/Year	Instructor	URL
Spring 2012	Dr. Läufer	https://luc.box.com/s/cq1dbfvyb6pej9b205oxtz67he1h93et

5.20 COMP 443: Computer Networks

5.20.1 Credit Hours

3

5.20.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

5.20.3 Description

This course surveys the construction of modern packet-switched networks, focusing on TCP/IP. At the LAN layer, protocols such as Ethernet and Wi-Fi are described. At the IP layer, a variety of routing-update protocols are studied. At the Transport layer, the course addresses the creation of end-to-end reliability and the management of network-wide congestion. The course also introduces basic issues in network security and in real-time communications. Some programming, generally in a language like Java or C#, may be required.

5.20.4 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Dordal	https://luc.box.com/s/cdcbz3v00yoebnsnh67bk1pq9ip02y01

5.21 COMP 446: Telecommunications

5.21.1 Credit Hours

3

5.21.2 Prerequisites

COMP 264: Introduction to Computer Systems and *COMP 271: Data Structures*

5.21.3 Description

This course introduces the fundamental concepts of telecommunication networks. Underlying engineering principles of telephone networks, computer networks and integrated digital networks are discussed. Topics in the course include: telephone and data networks overview; OSI layers; data link protocol; flow control, congestion control, routing; local area networks (Ethernet, Token Ring and FDDI); transport.

5.21.4 Syllabi

Semester/Year	Instructor	URL
Spring 2013	Dr. Dordal	https://luc.box.com/s/pvw81deb39p4d3h084iqr7y4tu3fflum

5.22 COMP 447: Intrusion Detection

5.22.1 Credit Hours

3

5.22.2 Prerequisites

COMP 271: Data Structures

5.22.3 Description

This course will cover techniques for detecting the unusual usage patterns that typically signal a break-in. The course will also consider differences in detection of local intruders versus intrusion over networks. Finally issues in the prosecution of those breaking in to computers, particularly evidentiary issues are explored.

5.22.4 Outcome

Students will learn to configure ID systems (eg., snort) and analyze their output. They will also understand both network-based and host-based monitoring techniques.

5.22.5 Syllabi

Semester/Year	Instructor	URL
Fall 2013	Corby Schmitz	https://luc.box.com/s/jwzbt7xvngsw1h7atoqwbnzs21kjry9s

5.23 COMP 448: Network Security

5.23.1 Credit Hours

3

5.23.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 170: Introduction to Object-Oriented Programming* or *COMP 447: Intrusion Detection*

5.23.3 Description

This course will involve a discussion of the methods and tactics used to keep attackers at bay as well as the mechanisms by which we can identify and potentially stop potential "bad guys." It will involve the following topics as they all relate to the overall security posture which makes computing safer: Encryption, authentication, firewalls, NAT/PAT, restricted access policies, intrusion detection and other security frameworks. The goal is to gain an understanding of how to secure computers and computing environments. Instructor permission is based on relevant industry experience with networking including TCP/IP.

5.23.4 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Corby Schmitz	https://luc.box.com/s/rjw2mw8iwtvdtt4jftpqvre09t78lrfe

5.24 COMP 449: Wireless Networks and Security

5.24.1 Credit Hours

3

5.24.2 Prerequisites

COMP 343: Introduction to Computer Networks or instructor permission

5.24.3 Description

In a mobile world, the ability to gain network access in a convenient manner, but yet securely, is becoming more and more of a requirement. This course will explore the wireless standards, authentication issues, common configuration models for commercial versus institution installs and analyze the security concerns associated with ad-hoc and standards-based methods of networking.

5.24.4 Syllabi

Semester/Year	Instructor	URL
Summer 2013	Corby Schmitz	https://luc.box.com/s/7gcetvnkj7sjfa4a7aw5puk1cf3mz82t

5.25 COMP 450: Microcomputer Design and Interfacing

5.25.1 Cross-Listing

Also Offered PHYS 366

5.25.2 Credit Hours

3

5.25.3 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures* or PHYS 266

5.25.4 Description

This course includes lecture and laboratory work. Topics that the course will hit on are; Microprocessor logic, instruction, and clocked sequential circuits; memory devices; data acquisition, manipulation and transfer circuitry; machine and assembly languages; microprocessor support devices.

5.25.5 Syllabi

Semester/Year	Instructor	URL
Fall 2014	William Honig	https://luc.box.com/s/xqid0wvx7inxqb8zv0543prstan192tj

5.26 COMP 451: Network Management

5.26.1 Credit Hours

3

5.26.2 Prerequisites

COMP 264: Introduction to Computer Systems or *COMP 271: Data Structures*

5.26.3 Description

Networks today are high-speed, heterogeneous, large-scale and delivers different media including data, audio and video. How do you effectively manage today's complex computer networks? This class provides complete yet accessible answers to network managers and researchers in this field. The course covers the basics of network management, alternative architectures, evaluation techniques, network management system components, SNMP and CMIP management protocols and the ISO network management applications: fault management, performance management, configuration management, security management, and accounting management. The course emphasizes the practical experience of developing network and distributed systems management tools using the SNMP++ and AdventNet wrappers. This course also highlights the latest advances in networks and distributed management area and shows case studies of academic and industrial systems such as HiFi, SMARRT, OpenView, NetView and Tivoli.

5.26.4 Outcome

Students will become familiar with the SNMP protocol, with how large-scale Network Management Systems operate and are configured, and with advanced network configuration.

5.26.5 Syllabi

Semester/Year	Instructor	URL
Spring 2011	Dr. Dordal	https://luc.box.com/s/zpggggqekdguy6bg25ocfctkwz8xzxny

5.27 COMP 453: Database Programming

5.27.1 Credit Hours

3

5.27.2 Prerequisites

COMP 271: Data Structures

5.27.3 Description

This course covers the fundamentals of database application development using C++, C, or Java by accessing a transaction-oriented database server. A commercial database environment such as Oracle is used. Apitional topics may include enabling access to database via the web, and administering large databases.

5.27.4 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Channah Naiman	https://luc.box.com/s/n47wvatvr9e453i9wh8o7av4kbzd2n8e

5.28 COMP 460: Algorithms and Complexity

5.28.1 Credit Hours

3

5.28.2 Prerequisites

COMP 363: Design and Analysis of Computer Algorithms

5.28.3 Description

The design and analysis of algorithms is central to computer science. This course will focus both on presenting general techniques for designing correct and efficient algorithms, as well as on formal methods for proving the correctness and analyzing the complexity of such algorithms. Also included will be an introduction to the theory of NP-completeness, whereby certain computation problems can be classified as being difficult in a formal sense.

5.28.4 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Sekharan	https://luc.box.com/s/8o8yme4d7ng3vdufzc6funme5clhx5i1

5.29 COMP 462: Computer Architecture

5.29.1 Credit Hours

3

5.29.2 Prerequisites

COMP 264: Introduction to Computer Systems or Comp 360 or comparable background, including but not limited to the following:

- Understanding of basic computer organization, including familiarity with such components as CPU, ALU, multiplexors, registers, main memory, caches, and buses

- familiarity with the roles of compilers, assemblers, and operating systems

- some familiarity with assembly language

- ability to understand simple C programs and to run programs in a UNIX environment, and

- familiarity with the representation of numbers in digital computers

5.29.3 Description

This course presents key principles underlying the design of modern digital computers. The course introduces quantitative techniques used to guide the design process. It describes CPU performance issues and introduces instruction set architectures. The course then uses a hypothetical computer design, with a simple RISC architecture, to show how modern digital computers are implemented, first using a simple non-pipelined implementation, followed by a higher-performace pipelined implementation. The major hazards introduced by pipelining, including structural hazards, data hazards, and control hazards are discussed and techniques for overcoming them are described. Additional topics covered in this course include the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

5.29.4 Outcome

Students gain an understanding of the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

5.29.5 Syllabi

No recent syllabi available.

5.30 COMP 464: High-Performance Computing

5.30.1 Credit Hours

3

5.30.2 Prerequisites

COMP 271: Data Structures

5.30.3 Description

This course is motivated by the realization that cpu speeds are levelling off at between 3GHz and 5GHz. The need for programs to be faster and more efficient is all the more important at a time when applications are getting increasingly larger and more complex. We will introduce techniques to gain performance boost in Java programs and C++ (or C) programs by discussing the use of multiple processors. This course will use two kinds of hardware platforms we have in the department, two IBM symmetric multiprocessors with 4 CPUs each and 4 GB of memory and a cluster of AMD machines all running Linux. We will study shared memory (OpenMP), message passing (MPI) and hybrid models of programming the high-performance machines. This course will use a blend of foundational understanding as well as a set of practical tools to gain insight into performance engineering of software.

5.30.4 Syllabi

Semester/Year	Instructor	URL
Fall 2015	Dr. Christopher P Stone	https://luc.box.com/s/z3gsxdhg8fjer6473yy69td4yc69y8d5

5.31 COMP 471: Programming Languages

5.31.1 Credit Hours

3

5.31.2 Prerequisites

COMP 413: Intermediate Object-Oriented Development

5.31.3 Description

There are over two thousand programming languages. This course studies several languages that represent the much smaller number of underlying principles and paradigms.

5.31.4 Outcome

An understanding of key principles and paradigms underlying the design and implementation of commonly used programming languages; exposure to formal mechanisms for describing language syntax and semantics; programming experience in several representative languages.

5.31.5 Syllabi

Semester/Year	Instructor	URL
Spring 2014	Dr. Läufer	https://luc.box.com/s/3pxq9mmize9n0pu9gusjke3y02vy4sp0

5.32 COMP 472: Compiler Construction

This course covers the basics of writing a compiler to translate from a simple high-level language to machine code. Topics include lexical analysis, top-down and LR parsing, syntax-directed translation, and code generation and optimization. Students will write a small compiler.

5.32.1 Credit Hours

3

5.32.2 Prerequisites

COMP 264: Introduction to Computer Systems and *COMP 313: Intermediate Object-Oriented Development*

5.32.3 Description

This course covers the basics of writing a compiler to translate from a simple high-level language to machine code. Topics include lexical analysis, top-down and LR parsing, syntax-directed translation, and code generation and optimization. Students will write a small compiler.

5.32.4 Outcome

Students will learn the theory and practice of how to build a compiler.

5.32.5 Syllabi

No recent syllabi available.

5.33 COMP 473: Object-Oriented Programming

5.33.1 Credit Hours

3

5.33.2 Prerequisites

COMP 413: Intermediate Object-Oriented Development or *COMP 313: Intermediate Object-Oriented Development*

You need to have knowledge of Java, C# or a similar object-oriented language, data structures, and fair sophistication with OOP - some experience with designing interacting, cooperating classes that would come in an intermediate discussion of software patterns. Software engineering is recommended but not required like the background listed above.

5.33.3 Description

Object-oriented programming has become the dominant software development paradigm of the 1990s. This course assumes familiarity with the main constituents of the object-oriented methodology:

- OOP = abstraction + inheritance + polymorphism

Building on this foundation, the course studies three major topics:

Design Patterns

First, the course discusses patterns, reusable solutions to recurring software design problems. Common design patterns such as Composite, Decorator, and Command typically involve several classes related by composition and inheritance.

Implementation

Next, the course examines how object-oriented language features such as composition, encapsulation, inheritance, and dynamic binding can be implemented. These features are studied in the context of an interpreter for a simplified object-oriented language.

Components

Finally, the course explores programming with reusable components. Specifically, the standard features of component architectures are discussed: properties, events, methods, and persistence, and introspection.

Other Information

This course is heavily project-oriented. Students study each major topic in the context of a substantial programming project. The reuse of existing class libraries and software components is emphasized. Grading will be based upon the following components: programming projects, quizzes, in-class exams, and the final exam.

5.33.4 Syllabi

Semester/Year	Instructor	URL
Fall 2014	Berhane Zewdie	https://luc.box.com/s/p1yy4kkiswuzeo2t3tk96qrfygnjolck

5.34 COMP 474: Software Engineering

5.34.1 Credit Hours

3

5.34.2 Prerequisites

COMP 271: Data Structures

5.34.3 Description

Using an object-oriented language such as C++ or Java, the student (working in a small team) will learn to plan, design, implement, and test a large software project.

5.34.4 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Christopher P Stone	https://luc.box.com/s/y8asvclz1yvlkyvhbbuuz8lypoby861x

5.35 COMP 475: System Standards and Requirements

5.35.1 Credit Hours

3

5.35.2 Prerequisites

None

5.35.3 Description

Introduction to the standards creation processes used by the computing and communications industry to generate technical and product standards. These standards influence world-wide growth of technology and often determine market success of companies. The course covers the structure and purpose of key formal and ad hoc standards groups across the world, explores the steps from the recognition of the need to the formal standard approval, introduces the processes, mechanisms, and technologies used to create standards, and gives special emphasis to the impact of the Open Source movement on software standards.

5.35.4 Syllabi

No recent syllabi available.

5.36 COMP 476: Formal Languages and Automata

5.36.1 Credit Hours

3

5.36.2 Prerequisites

COMP 163: Discrete Structures or MATH 201: Elementary Number Theory or MATH 212: Linear Algebra

5.36.3 Description

This course will study three mutually related topics: languages, machines, and computability. The mathematical ideas developed in this course are useful in many areas of computer science, including the design and specification of programming languages, construction of compilers, and exploring the capabilities and limitations of mechanical computation. This subject is important for the scientific foundations it lays for computer science, for the philosophical concerns it raises about the nature of computation, and for the sheer elegance it brings in to the studies related to a

variety of applications. Some of the most fundamental discoveries in computer science identify connections among languages, machines, and computability. Furthermore, some of the most challenging questions at the heart of computer science also arise from these topics. The course will cover a majority of the following topics: regular languages, finite automata, determinism and nondeterminism in finite automata, applications to searching and pattern matching, context-free languages, push-down automata, applications to compiler design, computability theory, Church-Turing thesis, Turing machines, undecidability, recursive and recursively enumerable languages, reductions among languages, resource-bounded computation, Kolmogorov complexity.

5.36.4 Syllabi

Semester/Year	Instructor	URL
Fall 2014	Stephen Doty	https://luc.box.com/s/94r7s06j61y0ufprujo0bm7147ek63qz

5.37 COMP 477: IT Project Management

5.37.1 Credit Hours

3

5.37.2 Prerequisites

COMP 271: Data Structures

5.37.3 Description

This course is an introduction to the philosophy and practice of project management. The course involves a student group project to investigate and plan a "real world" project. The investigation requires application of project-management tools covered in the class, including a project proposal that specifies project objectives, schedules, work breakdown structure, and responsibilities, an written interim report, and a final oral and written report. The course will likely include both business and computer science students working together on a student team.

5.37.4 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Dr. Channah Naiman	https://luc.box.com/s/1tgu3y4lxl7uy2kc2mct0wyo2iu4ls7y

5.38 COMP 480: Operations Management

5.38.1 Credit Hours

3

5.38.2 Prerequisites

COMP 271: Data Structures

5.38.3 Description

Introduction to concepts and methods for managing production and service operations. Topics include demand forecasting, aggregate and capacity planning, inventory management, facility layout and location, just-in-time, managing quality, project planning, resource allocation, and logistics. Emphasis on models to support decision making.

Outcome: Understanding of role of operations management in organizations, and the ability to use the modleing tools and techniques for problem-solving in operations management.

5.39 COMP 484: Artificial Intelligence

5.39.1 Prerequisites

COMP 271: Data Structures

5.39.2 Description

This course provides an introduction into the field of artificial intelligence (AI). Topics include search, game-playing, logic, planning, uncertainty, learning, and perception, as time permits.

5.39.3 Other Information

There will be a midterm and final exam and some programming assignments.

5.39.4 Syllabi

Semester/Year	Instructor	URL
Spring 2016	Conrad Weisert	https://luc.box.com/s/btamshxy6n0g5wme57ahthbefyt16d67

5.40 COMP 488: Topics in Computer Science

5.40.1 Credit Hours

3

5.40.2 Prerequisites

Vary by section: most commonly *COMP 271: Data Structures*

5.40.3 Description

This course is used to introduce emerging topics in computer science that do not yet have a regular course number. Content of the course varies. Recent topics have included

- Bioinformatics
- Computational Biology

- Computational Neuroscience

- Enterprise Software Development

- Introduction to Digital Humanities Design and Programming

- Machine Learning

- Metagenomics

- Rapid Application Development Metholodgy (.NET)

- Robotics Software Development

5.40.4 Outcome

Understand an emerging area of Computer Science.

5.40.5 Syllabi

Topic	Semester/Year	Instructor	URL
Physical Design and Fabrication	Spring 2016	Jonathan Durston	https://luc.box.com/s/ 2oyte72cslhh13feckwby10ujwz198th
Introduction to Computer Forensics	Spring 2016	Thomas Yarish	https://luc.box.com/s/ sa7ybkvrd3elmf7yj7voxqgt4dgoqg30
Data Warehousing and Data Mining	Spring 2016	Dr. Channah Naiman	https://luc.box.com/s/ f82hzdyad9felysw5ual3l9pm7mn6apw
Metagenomics	Fall 2015	Dr. Putonti	https://luc.box.com/s/ bg4vgtlumwnxafuq1tdvwf33mhvfw5a6
Introduction to Digital Humanites Design and Programming	Spring 2015	Nicholas Hayward	https://luc.box.com/s/ q4k5x000vydb9av4lb3z2p6wr6f8wps1
Software Quality and Testing	Fall 2014	Conrad Weisert	https://luc.box.com/s/ yrm6n3065hbbsbv4khq3tlvm47j93cf5
Organizational Change & Development	Spring 2017	Guy Bevente	coming; see below now
Past Offerings			https://luc.box.com/s/ ldp6gn86hexwotsvdm3e0q5rifidk8np

Organizational Change and Development (syllabus coming later)

This course focuses on the manager's role in leading Organization Development and Change to maximize organization and individual effectiveness with a focus on Information Technology challenges. The class will explore the theory of Organization Development and Change, introduce Change Management practices and methods to facilitate change, and discuss operational, employee, and financial considerations a manager will face as a change agent.

The course will introduce IT specific use cases such as Outsourcing, Build versus Buy, Hosting, and building an Innovation culture. The class goal is to enhance the leader's effectiveness in leading change by applying proven techniques, assessing issues that will surface, understanding the forces of resistance, and implementing decisions using frameworks that lead to informed management action.

5.41 COMP 490: Independent Project

5.41.1 Credit Hours

Variable (1-6)

5.41.2 Prerequisites

Previous agreement of a Computer Science faculty supervisor

5.41.3 Description

An independent project in computer science or related disciplines, under the supervision of a member of faculty.

Students may not register themselves. A request with the exact semester and number of units must be forwarded by the faculty supervisor, after an agreement is made with the student.

A good feature of an independent project is that you can make your transcript show a course title that is specific to your particular project. Fill out the Request for Course Title form, and get the signature of your supervisor and the gpd. More information can be found in the Graduate Program Handbook.

5.42 COMP 499: Internship

5.42.1 Credit Hours

Variable (1-6)

5.42.2 Prerequisites

Agreement of the Graduate Program Director (GPD)

5.42.3 Description

An opportunity to obtain experience in software development, design, networks, or related activities in computer science in a professional setting. The student must obtain the approval of the Graduate Program Director and the student's work supervisor. A final report from the student and the supervisor are required. More information can be found in the Graduate Program Handbook.

5.43 COMP 605: Master of Science Study

5.43.1 Credit Hours

0

5.43.2 Prerequisites

Must be a Student in the M.S. Computer Science Program and doing the Thesis Option.

5.43.3 Description

This course is only for students who are actively working on their thesis, but not taking enough regular courses to be full-time. Enrolling in this course will give you full-time student status.

DOWNLOADING

- downloading

genindex