CS 200 - Programming I

Marc Renault

Department of Computer Sciences
University of Wisconsin – Madison

Spring 2018

TopHat Sec 3 (AM) Join Code: 427811
TopHat Sec 4 (PM) Join Code: 165455

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

CS 200 - Programming I: Spring 2018

Computer Science and Programming

Computer Science

Programming

Computer Science and Programming

Computer Science

- Broad discipline that explores any and all areas of computation.
- Includes: theory of computation, algorithms, computer graphics, language theory, systems, and human-computer interaction.

Programming

Computer Science and Programming

Computer Science

- Broad discipline that explores any and all areas of computation.
- Includes: theory of computation, algorithms, computer graphics, language theory, systems, and human-computer interaction.

Programming

- Providing a set of instructions to a computer to automate a specific task or solve a given problem.

Computer Science and Programming

Computer Science

- Broad discipline that explores any and all areas of computation.
- Includes: theory of computation, algorithms, computer graphics, language theory, systems, and human-computer interaction.

Programming

- Providing a set of instructions to a computer to automate a specific task or solve a given problem.
- An application of computer science.

Computer Science and Programming

Computer Science

- Broad discipline that explores any and all areas of computation.
- Includes: theory of computation, algorithms, computer graphics, language theory, systems, and human-computer interaction.

Programming

- Providing a set of instructions to a computer to automate a specific task or solve a given problem.
- An application of computer science.
- Implementing algorithms.

CS 200 Emphasises Declarative and Procedural Knowledge

Declarative Knowledge

- Knowing that something is the case.

CS 200 Emphasises Declarative and Procedural Knowledge

Declarative Knowledge

- Knowing that something is the case.
- Knowing the CS jargon.
- Knowing the Java syntax.

CS 200 Emphasises Declarative and Procedural Knowledge

Declarative Knowledge

- Knowing that something is the case.
- Knowing the CS jargon.
- Knowing the Java syntax.

Procedural Knowledge

- Knowing how to do something.

CS 200 Emphasises Declarative and Procedural Knowledge

Declarative Knowledge

- Knowing that something is the case.
- Knowing the CS jargon.
- Knowing the Java syntax.

Procedural Knowledge

- Knowing how to do something.
- Solving a problem systematically.
- Writing that solution in Java.

About You

My current year in school is:

a. Freshman
b. Sophomore
c. Junior
d. Senior
e. Graduate Student
f. Other

About You

My primary reason for taking CS 200:

a. I am very interested in the subject.
b. I am curious to learn more about the subject.
c. It fulfils a requirement for my CS major or certificate.
d. It fulfils a requirement outside of the CS major or certificate.
e. It fits my schedule.
f. I've heard good things about the course.

About You

I have daily access (outside of university labs) to computers with the following operating systems:

a. Windows
b. Mac
c. Linux
d. Other
e. None

About You

My favourite Star Wars movie (from the trilogies) is:

a. I - The Phantom Menace
b. II - Attack of the Clones
c. III - Revenge of the Sith
d. IV - A New Hope
e. V - The Empire Strikes Back
f. VI - Return of the Jedi
g. VII - The Force Awakens
h. VIII - The Last Jedi
i. N/A - Never seen them

CS 200 Programming I

Spring 2018 Team

https://cs200-www.cs.wisc.edu/wp/contact/

Instructors

Jim Williams
jimw @cs.wisc.edu
Lectures 001 and 002
Office: 6384 CS
Hours: T 2:3Opm - 4:30 pm, W 9:30am - 11:30am, or by appt.

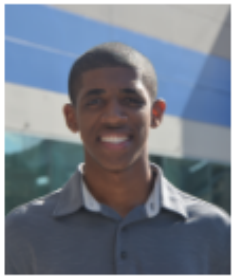
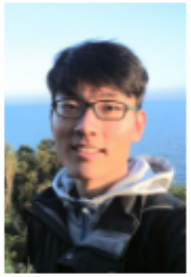
Marc Renault
mrenault @cs.wisc.edu
Lectures 003 and 004
Office: 6382 CS
Hours: T 12:30pm - 2:30pm, Th 1:00pm - 3:00pm, or by appt.

CS 200 Programming I

Spring 2018 Team

https://cs200-www.cs.wisc.edu/wp/contact/

Consultants (Teaching and Lab Assistants)

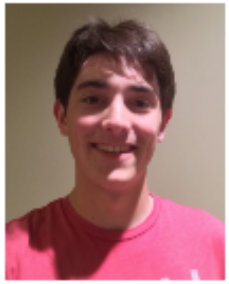
Yuan Hsieh  yhsieh28 @WISC.EDU	**Ajene** Johnson  johnson48 @WISC.EDU	**Shantanu** Gupta  sgupta226 @WISC.EDU	**Jack** Truskowski  jtruskowski @WISC.EDU
Jaya Rangarajan  jrangarajan @WISC.EDU	**Varun** Ramesh  varunr @CS.WISC.EDU	**Yuanfang** Song  ysong243 @WISC.EDU	**Henry** Lv  hlv2 @WISC.EDU

CS 200 Programming I

Spring 2018 Team

https://cs200-www.cs.wisc.edu/wp/contact/

Consultants (Teaching and Lab Assistants)

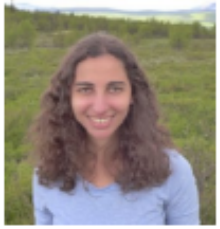
Srujith Poondla  spoondla @WISC.EDU	**Maggie** Marxen  mmarxen @WISC.EDU	**David** Neiman  dneiman @WISC.EDU	**Grace** Johnson  gejohnson2 @WISC.EDU
Sherine Zhang  xzhang662 @WISC.EDU	**Ray** Zhang  rmzhang @WISC.EDU	**Michael** Guyse  guyse @WISC.EDU	**Ben** Schulman  bgschulma @WISC.EDU

CS 200 Programming I

Spring 2018 Team

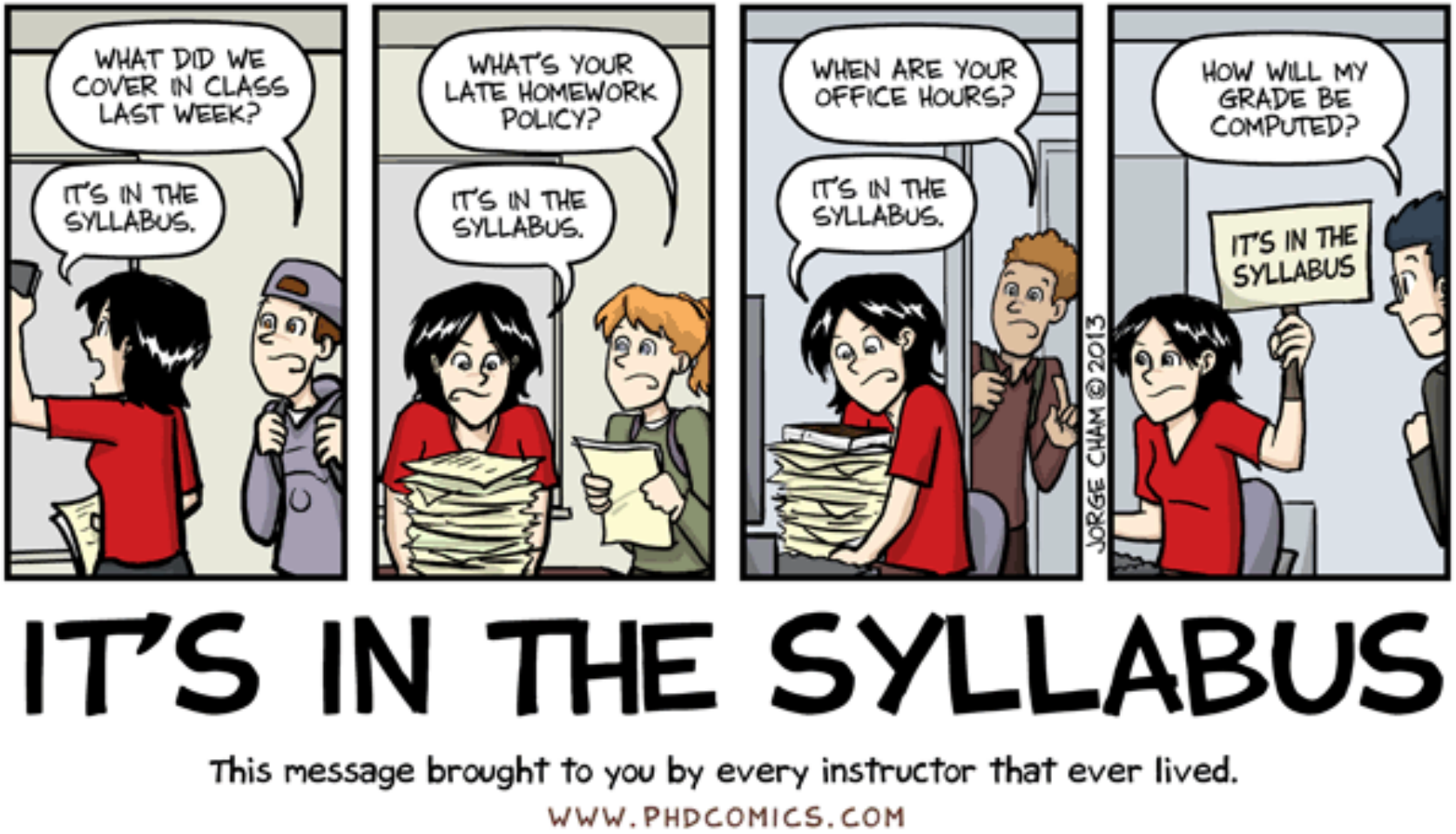
https://cs200-www.cs.wisc.edu/wp/contact/

Consultants (Teaching and Lab Assistants)

Mark Wiemer  mwiemer2 @WISC.EDU	**Hadar** Ohana  hohana @WISC.EDU	**Mike** Nedbai  nedbai @WISC.EDU	**Connor** Waity cbwaity @WISC.EDU
Jake DeCamp  decamp2 @WISC.EDU			

CS 200 Website

https://cs200-www.cs.wisc.edu/



Course Aim

https://cs200-www.cs.wisc.edu/wp/syllabus/

Overall

- Intended for students who have no prior programming experience.
- Teach the process of incrementally developing small programs along with fundamental CS topics.
- Key topics: problem abstraction, edit-compile-run cycle, data types, control structures, basic testing and debugging, and good programming practices.

Course Aim

https://cs200-www.cs.wisc.edu/wp/syllabus/

Specific Learning Outcomes

- Design and implement a standalone program that can interact with the user via prompts and or menus, access and edit data stored in an array or list structure, and use and further process the data found in those structures.

Course Aim

https://cs200-www.cs.wisc.edu/wp/syllabus/

Specific Learning Outcomes

- Design and implement a standalone program that can interact with the user via prompts and or menus, access and edit data stored in an array or list structure, and use and further process the data found in those structures.
- Able to trace code to determine output or results.

Course Aim

https://cs200-www.cs.wisc.edu/wp/syllabus/

Specific Learning Outcomes

- Design and implement a standalone program that can interact with the user via prompts and or menus, access and edit data stored in an array or list structure, and use and further process the data found in those structures.
- Able to trace code to determine output or results.
- Able to implement a given program design and choose correct control structures for implementing algorithms expressed in pseudocode.

Course Aim

https://cs200-www.cs.wisc.edu/wp/syllabus/

Specific Learning Outcomes

- Able to interpret a variety of diagram types used to express programming concepts and results: truth tables, memory model diagrams, control flow charts, class diagrams, object diagrams, and use-case diagrams.

Course Aim

https://cs200-www.cs.wisc.edu/wp/syllabus/

Specific Learning Outcomes

- Able to interpret a variety of diagram types used to express programming concepts and results: truth tables, memory model diagrams, control flow charts, class diagrams, object diagrams, and use-case diagrams.
- List, describe, use the basic I/O operations for reading and writing text files to and from the computer's hard drive.

Getting Started

Getting Started

https://cs200-www.cs.wisc.edu/wp/

Checklist

1. Best Course for you?
2. zyBook Registration
3. TopHat Registration
4. Activate Piazza account
5. Activate CS Account
6. Find Team Lab
7. Review the Syllabus
8. By week 3: Exam conflicts and accommodations
9. By week 3: Install Java 8 and Eclipse on your computer

1. Best Course for you?

New to Programming

- CS 200 is intended for CS majors or those who are planning on doing more CS courses.
- CS 301 is intended for non-CS majors and is taught using Python.

1. Best Course for you?

New to Programming

- CS 200 is intended for CS majors or those who are planning on doing more CS courses.
- CS 301 is intended for non-CS majors and is taught using Python.

Experience in Programming

- Try the first assignment from CS 300 (`http://cs300-www.cs.wisc.edu/wp/`).
- If you can do it without help, talk to CS 300 instructor about enrolling.

2. zyBook Registration

Required:

COMP SCI 200: Programming I
zyBooks.com, 2015.
zyBook code:
WISCCOMPSCI200Spring2018

2. zyBook Registration

Required:

COMP SCI 200: Programming I
zyBooks.com, 2015.
zyBook code:
WISCCOMPSCI200Spring2018

Use zyBooks for

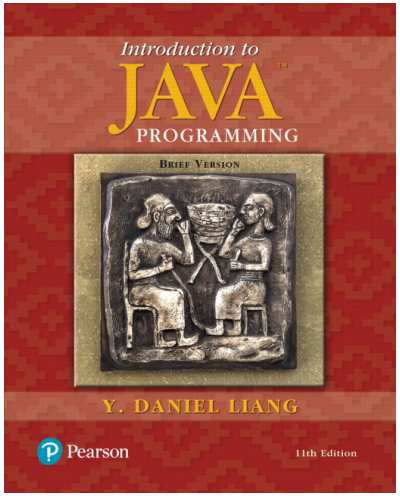
- Textbook
- Participation activities (Before lecture - 5%)
- Challenge activities (After lecture - 5%)
- Programming Assignments (Lab activities)

2. zyBook Registration

Required:

COMP SCI 200: Programming I
zyBooks.com, 2015.
zyBook code:
WISCCOMPSCI200Spring2018

Additional reference:



Liand, Daniel Y. ***Introduction to Java Programming, Brief Version, 11th Edition.*** **Pearson, 2018.**

3. TopHat Registration

TOP HAT

Join Codes:

- Lecture 003 MWF (AM): 427811
- Lecture 004 MWF (PM): 165455

3. TopHat Registration

TOP HAT

Join Codes:

- Lecture 003 MWF (AM): 427811
- Lecture 004 MWF (PM): 165455

In-class participation

- Facility classroom participation.
- Participation grade (5%)
- Grade is calculated on a per class per week basis.

4. Activate Piazza Account

piazza

Online question resource

- One discussion area for all sections.
- Interaction of students, consultants and instructors.
- First stop for getting questions answered.

4. Activate Piazza Account

piazza

Online question resource

- One discussion area for all sections.
- Interaction of students, consultants and instructors.
- First stop for getting questions answered.

Rules

- Be courteous.
- Don't post code!
- Search first, post second.
- Read Piazza Expectations post.

5. Activate your CS Account

CS Account Key Points

- All registered CS 200 students are provided an account for the CS lab.
- The username and password can differ from your UW NetID.
- Some accounts may not be available until after the first day of classes.
- 1 day delay in activation.
- Needed to log into lab computers.

6. Prepare for Team Labs

Team Labs

- Start Week 2
- Review the previous week zyBooks chapter.
- Broken up into labs of 40 students with 1 consultant per 12 to 14 students.
- Pair programming.
- Labs are designed to be interactive.

6. Prepare for Team Labs

Team Labs

- Start Week 2
- Review the previous week zyBooks chapter.
- Broken up into labs of 40 students with 1 consultant per 12 to 14 students.
- Pair programming.
- Labs are designed to be interactive.
- Marks are participation based:
  - Arrive on time (within 5 minutes): 1 point
  - Leave on time (within 5 minutes): 1 point
  - Participation: 3 points

7. Review the Syllabus

Grading

- Participation (20%)
  - zyBooks Participation Activities (5%)
  - zyBooks Challenge Activities (5%)
  - Team Labs (5%)
  - TopHat Questions (5%)

7. Review the Syllabus

Grading

- Participation (20%)
  - zyBooks Participation Activities (5%)
  - zyBooks Challenge Activities (5%)
  - Team Labs (5%)
  - TopHat Questions (5%)
  - 80% rule

7. Review the Syllabus

Grading

- Participation (20%)
  - zyBooks Participation Activities (5%)
  - zyBooks Challenge Activities (5%)
  - Team Labs (5%)
  - TopHat Questions (5%)
  - 80% rule
- Exams (45%: $\max\{0.45 \cdot \text{E3}, 0.2 \cdot \text{E3} + 0.15 \cdot \text{E2} + 0.1 \cdot \text{E1}\}$)
  - Pre-exam 1 – Friday, Feb 16 in lecture
  - Exam 1 – Thursday, Mar 1 (5pm to 7pm)
    $\text{E1} := \max\{\text{Exam 1}, 0.8 \cdot \text{Exam 1} + 0.2 \cdot \text{Pre-exam 1}\}$
  - Exam 2 (E2) – Thursday, Apr 12 (5pm to 7pm)
  - Final Exam (E3) – Sunday, May 6 (7:45am to 9:45am)

7. Review the Syllabus

Grading

- Participation (20%)
  - zyBooks Participation Activities (5%)
  - zyBooks Challenge Activities (5%)
  - Team Labs (5%)
  - TopHat Questions (5%)
  - 80% rule
- Exams (45%: $\max\{0.45 \cdot \text{E3}, 0.2 \cdot \text{E3} + 0.15 \cdot \text{E2} + 0.1 \cdot \text{E1}\}$)
  - Pre-exam 1 – Friday, Feb 16 in lecture
  - Exam 1 – Thursday, Mar 1 (5pm to 7pm)
    $\text{E1} := \max\{\text{Exam 1}, 0.8 \cdot \text{Exam 1} + 0.2 \cdot \text{Pre-exam 1}\}$
  - Exam 2 (E2) – Thursday, Apr 12 (5pm to 7pm)
  - Final Exam (E3) – Sunday, May 6 (7:45am to 9:45am)
- Programs (35%)
  - Top 5 of 7 weekly programming assignments (P1 - P7) (15%)
  - 2 multi-week assignments (BP1 & BP2) (20%)

7. Review the Syllabus

No Late Assignments Accepted

- 80% rule for Participation items.
- In extreme circumstances, contact me ASAP and submit what has been done.

7. Review the Syllabus

No Late Assignments Accepted

- 80% rule for Participation items.
- In extreme circumstances, contact me ASAP and submit what has been done.

Academic Integrity

- Academic dishonesty or misconduct is taken very seriously by the university (see UW–Madison Academic Integrity policy).
- It is academic misconduct to submit someone else's work as your own.
- It is academic misconduct to help another student commit academic misconduct.

The Rest (by week 3)

8. Determine exam conflicts and other accommodations requests.

- Exam 1 – Thursday, Mar 1 (5pm to 7pm)
- Exam 2 – Thursday, Apr 12 (5pm to 7pm)
- Final Exam – Monday, May 6 (7:45am to 9:45am)

The Rest (by week 3)

8. Determine exam conflicts and other accommodations requests.

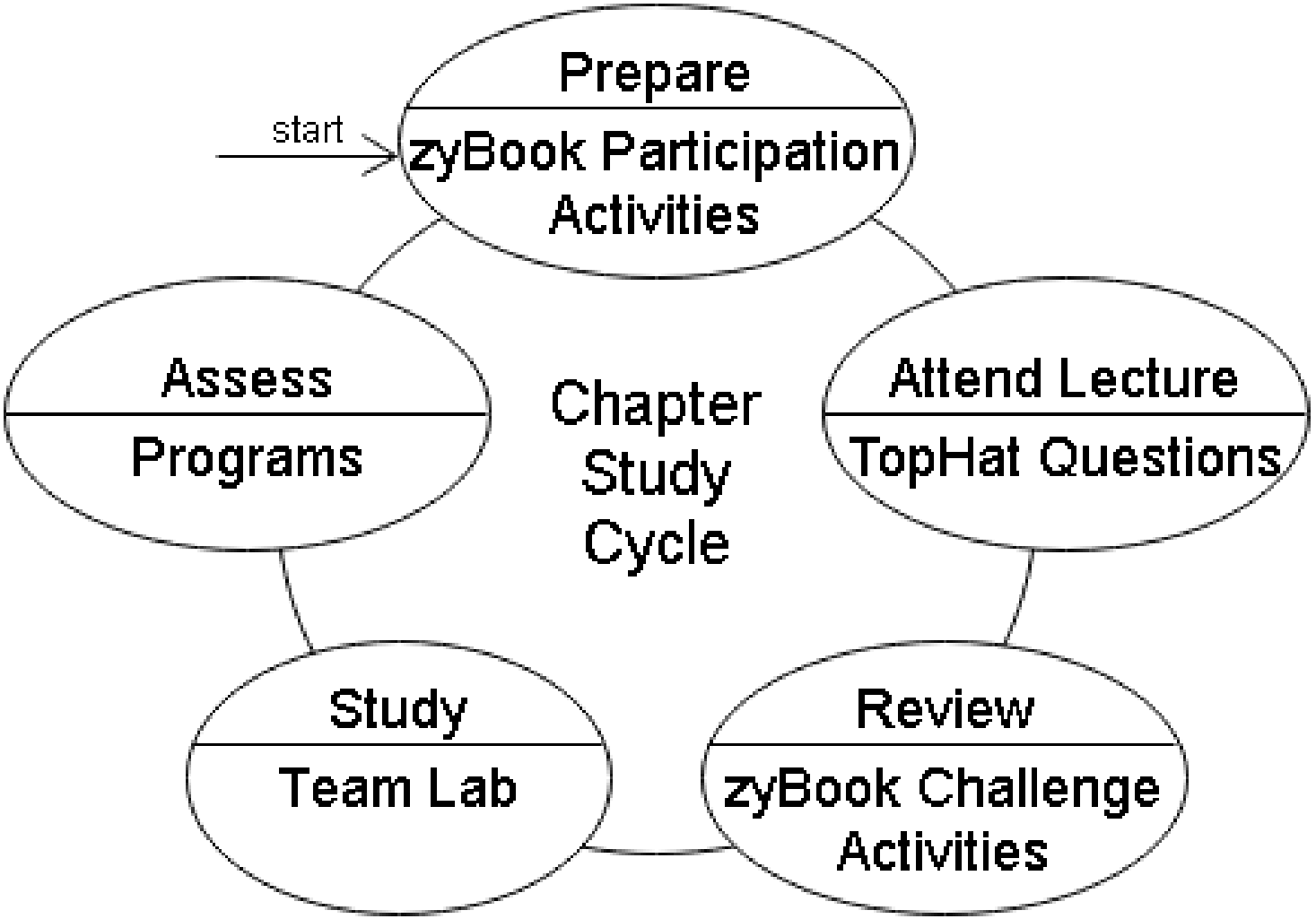
- Exam 1 – Thursday, Mar 1 (5pm to 7pm)
- Exam 2 – Thursday, Apr 12 (5pm to 7pm)
- Final Exam – Monday, May 6 (7:45am to 9:45am)

9. Install Java 8 and Eclipse on your computer.

- `https://cs200-www.cs.wisc.edu/wp/install-eclipse/`
- Consultants can help if you are having trouble with the installation.

Study Cycle

Study Cycle


start
Prepare
zyBook Participation Activities
Attend Lecture
TopHat Questions
Review
zyBook Challenge Activities
Study
Team Lab
Assess
Programs
Chapter Study Cycle

zyBooks Programs, BP1 and BP2

Keys to Success

- Start early: try getting started the weekend before.
- Focus on a systematic approach to the problem.
- Test your work!

zyBooks Programs, BP1 and BP2

Keys to Success

- Start early: try getting started the weekend before.
- Focus on a systematic approach to the problem.
- Test your work!

Unlimited submissions before Deadline

- Final submission is highest mark closest to the deadline.
- 30 minute delay between submissions.
- Not intended to be a test bench.

Getting Help

Getting Help

https://cs200-www.cs.wisc.edu/wp/help/

Help!

- Tips for Solving Programming Problems
- Piazza Online Discussion
- Consultants
  - MTWThF: 2:00pm to 5:30pm – 1358 CS
  - MTWTh: 5:30pm to 8:00pm – 1350 CS or 1370 CS
  - MF: Appointments possible.
- Instructor Office Hours
- CS Learning Center
- Drop-in Tutoring (College of Engineering)

A couple more questions.

Which of the following is the best estimate for the amount of computer code that you have written prior to this class?

a. 0 lines (I have never written any code)
b. 1 – 200 lines (I have written some code, but not a lot)
c. 201 – 2,000 lines (I have written several small programs)
d. 2,000 – 20,000 lines
e. $>$ 20,000 (I have done a substantial amount of programming)

A couple more questions.

I am expecting to spend approximately this many hours per week on this class:

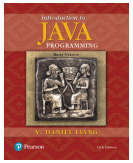
a. Less than 4 hours/week
b. 4 – 8 hours/week
c. 8 – 12 hours/week
d. 12 – 16 hours/week
e. More than 16 hours/week

Appendix

References

Image Sources I

https://cs200-www.cs.wisc.edu/wp/



https://www.pearson.com/us/higher-education/program/Liang-Introduction-to-Java-Programming-Brief-Ver PGM1730772.html

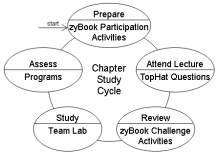
TOP HAT

https://tophat.com/

https://cs200-www.cs.wisc.edu/wp/

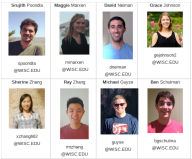
Image Sources II

https://piazza.com/



https://cs200-www.cs.wisc.edu/wp/syllabus/

https://cs200-www.cs.wisc.edu/wp/



https://cs200-www.cs.wisc.edu/wp/

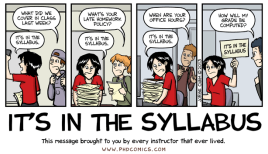
https://cs200-www.cs.wisc.edu/wp/

https://brand.wisc.edu/web/logos/

Image Sources III

http://www.zybooks.com/

http://bigpicture.typepad.com/comments/images/2008/07/14/dont_panic.png



http://phdcomics.com/comics.php?f=1583