MSCI 240: ALGORITHMS & DATA STRUCTURES

Instructor: Mark Hancock

lecture summary

administrivia

expectations

course objectives

what are data structures and algorithms?

clicker questions

administrivia

Instructor: Prof. Mark Hancock

office: CPH 3633

email: mark.hancock@uwaterloo.ca

phone: (519) 888-4567 ext. 36587

office hours: Mondays @ 11:30-12:30 (starts Sep 10)

TA: Aritra Mitra

- email: aritra.mitra@uwaterloo.ca
- office hours: Fridays @ 11:30-12:30

TA: Kourosh Khedriliraviasl

- email: kourosh.khedriliraviasl@uwaterloo.ca
- office hours: Tuesdays @ 11:30-12:30

TA: Ishita Goswami (half time)

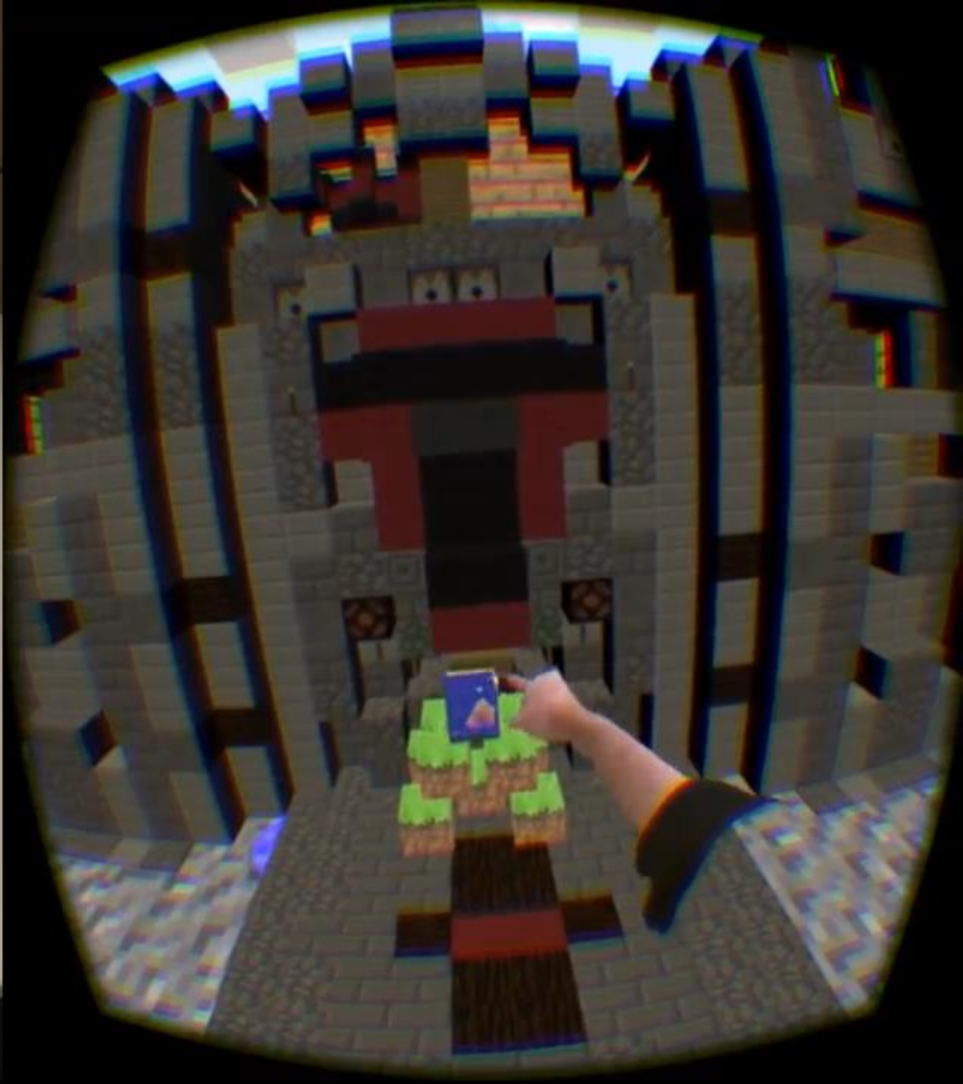
- email: igoswami@uwaterloo.ca
- office hours: Thursdays @ 11:30-12:30

who is Prof. Mark Hancock?

research areas:

- human-computer interaction
- novel interactive technology
- games research

us

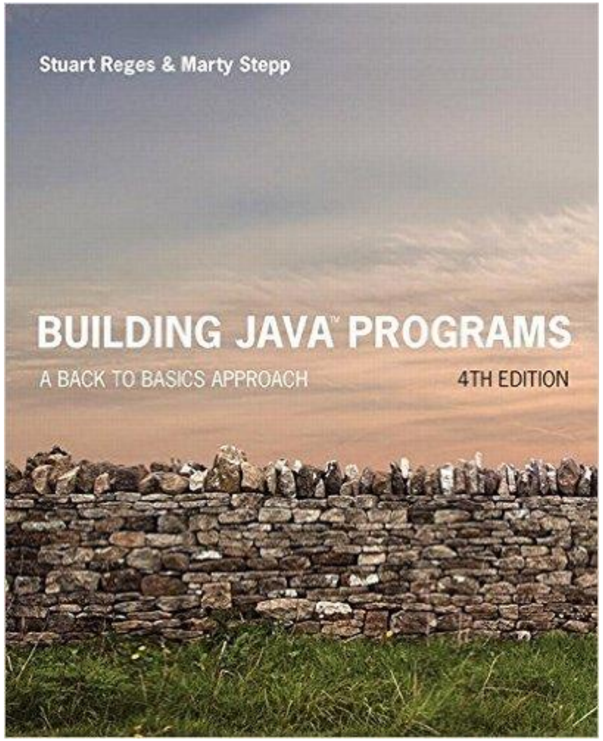


textbook one:

Building Java Programs: A Back to Basics Approach, 4th Edition, by Stuart Reges & Marty Stepp (2016).

this book is available from the university bookstore and through university library's course reserves

online resources: http://www.buildingjavaprograms.com/



textbook two:

Algorithms, 4th Edition, by Robert Sedgewick and Kevin Wayne (2011).

this book is available from the university bookstore and through university library's course reserves

online resources: http://algs4.cs.princeton.edu/home/

clickers/REEF:

you are required to have a clicker/REEF and bring it to all lectures and tutorials

you can purchase a clicker/REEF code from the bookstore

you must register your clicker through Learn, REEF instructions are also on Learn

make sure you’re on Learn!

all important information will be posted there:

- syllabus
- announcements
- assignments
- grades
- extra resources

grading:

Assignments (35%)	
Labs (4-6 marked labs)	15%
Homework (2-5 assignments)	
Tutorials/Sports Day/Clicker	
Programming Assignments/Projects (3-4, each worth ~equal amount)	20%
Exams/Quizzes (65%)	
Midterm	20%
Final Exam	45%

(tentative) midterm date:

Wednesday evening, Nov 7 (check with other courses, discuss Monday)

tutorials/labs:

start week of Sep 17 (none next week)

academic integrity (read the syllabus!):

- plagiarism
- cheating
- misrepresentation
- impersonation
- collaboration
- http://www.lib.uwaterloo.ca/ait/

what happens if suspected of cheating?

TA will report to me, I will report directly to the Faculty of Engineering Undergraduate Office with evidence

no warnings—this will be the first action taken

undergraduate office will carry out the investigation, will talk to you and ask you to justify why you cheated

you will get a 0 on the assignment, your overall grade will be reduced by 5% and there will be a letter in your file (if you have cheated before, there will be more severe consequences) bottom line: don’t cheat! it’s really not worth it!

if you’re unsure what counts as cheating, please come talk to us!

any of these acts are considered cheating:

- copying code or code fragments from any source
- copying even one line of source code
- copying even one line from some published source (web, book, etc.) without quoting it and providing a proper citation
- submitting code that has only minor modifications of another's work, e.g. changing variable names
- providing code or code fragments (by paper or electronically) to another student (or group)
- showing another student your assignment
- writing any part of another student's assignment (even one line of code) for them
- submitting output (screen capture) that was not generated by the code submitted
- entering clicker answers on behalf of another student

all work must be your own

you can work together

best practice for collaboration:

- work at whiteboards with others using pseudocode
- no one takes notes
- erase whiteboard when done

my expectations

professional attitude

- questions in class, email, group work

attend lectures, tutorials, and labs

- if missed, get notes from classmate

participation

- class discussions, activities, clickers
- tutorials, labs

course objectives

by the end of this course you will be able to…

1. understand and use abstraction to separate interface from implementation

2. analyze the performance of solutions:
   a. empirically measure and analyze the efficiency of a computer program
   b. analyze algorithms and predict their ability to scale with increasing input sizes

3. select appropriate data structures and algorithms for programming tasks

what is a data structure?

how do you store your socks?

how do you store your shirts?

how do you store your notes for class?

how are these different?

what operations do you need to be able to do?

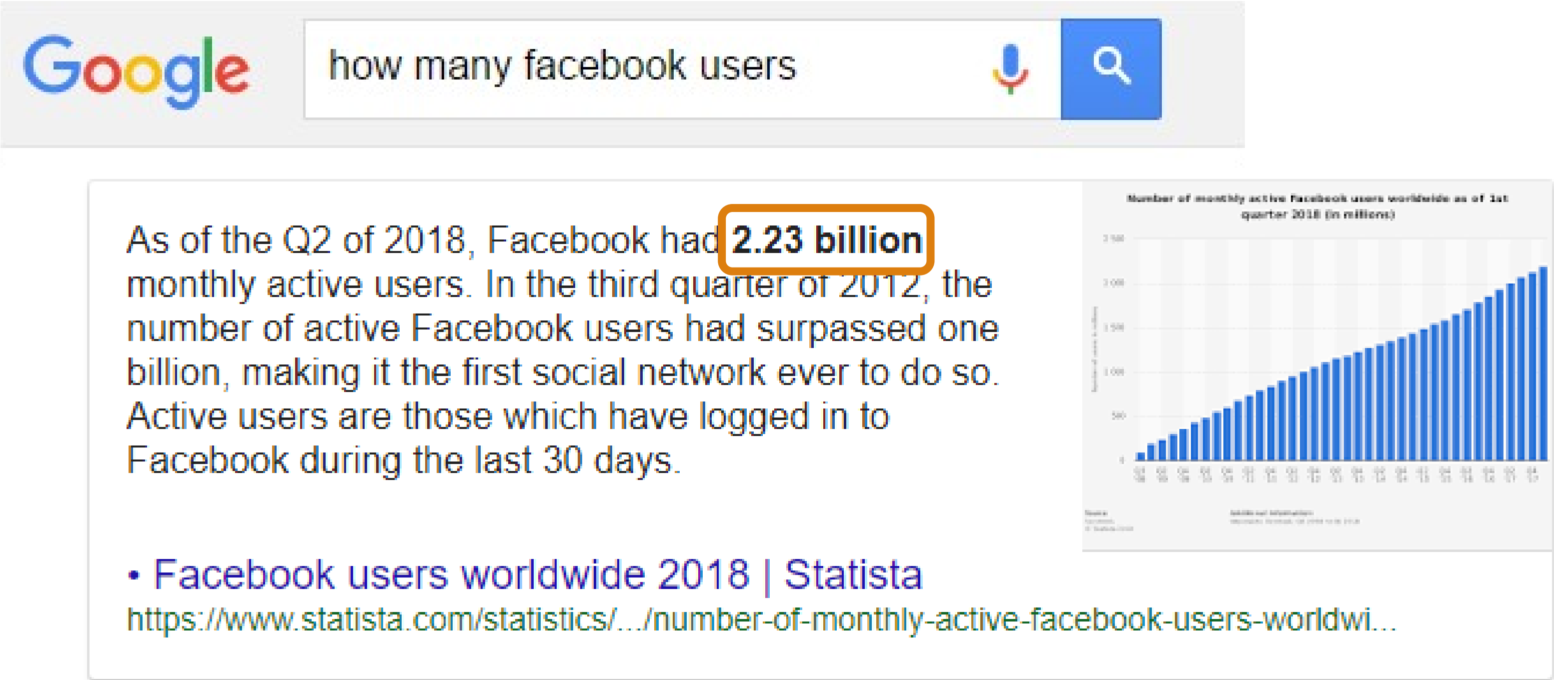
what is an algorithm?

MSCI 121:
an algorithm is a step-by-step procedure to solve a problem

more technically (now that you’ve taken 121): an algorithm is a program or method of manipulating/computing with data

e.g., sort a list of numbers in ascending order

why do they matter?


Google
how many facebook users
Number of monthly active Facebook users worldwide as of 1st quarter 2018 (in millions)
2 500
2 000
1 500
1 000
500
0

As of the Q2 of 2018, Facebook had **2.23 billion** monthly active users. In the third quarter of 2012, the number of active Facebook users had surpassed one billion, making it the first social network ever to do so. Active users are those which have logged in to Facebook during the last 30 days.

- Facebook users worldwide 2018 | Statista

https://www.statista.com/statistics/.../number-of-monthly-active-facebook-users-worldwi...

how do you store this much data?

what operations can you do on this much data?

how quickly can you access a single user’s data?

efficiency

- cost to build and maintain
- measurement: empirical & analytical

design

- pros/cons of different options
- know before you build it

clicker questions

how much of the whiteboard can you see?

A. top 20%

B. top 40%

C. top 60%

D. top 80%

E. all of the board (top 100%)

on co-op, I worked in:

A. information technology / software

B. consulting

C. manufacturing

D. construction

E. none of the above

as part of my co-op:

A. I wrote computer programs on a daily basis

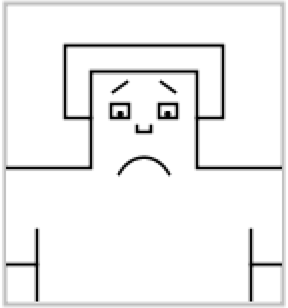
B. I wrote at least one computer program

C. I was involved with developing software, but I did not write any programs

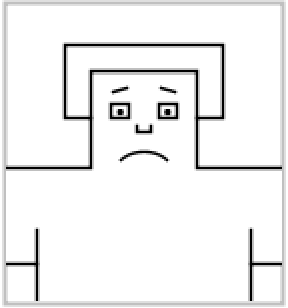
D. I did nothing related to software development

E. No answer

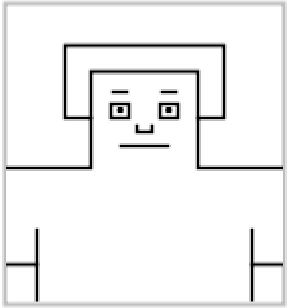
“Algorithms & Data Structures”:



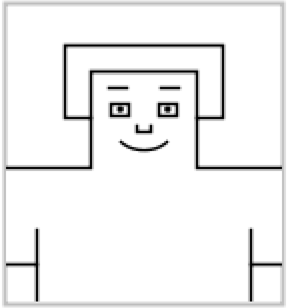
A



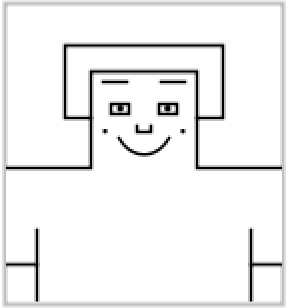
B



C

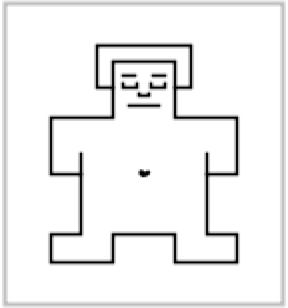


D

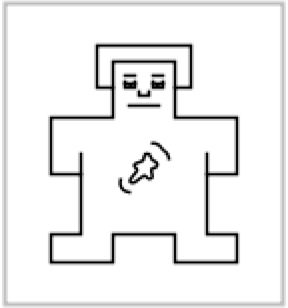


E

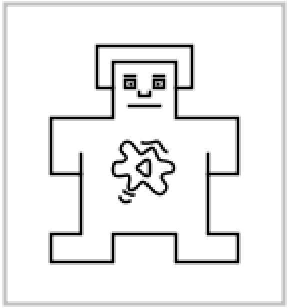
“Algorithms & Data Structures”:



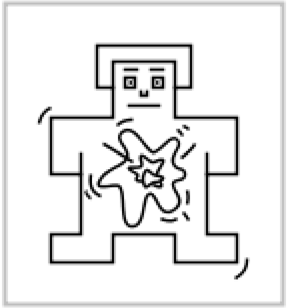
A



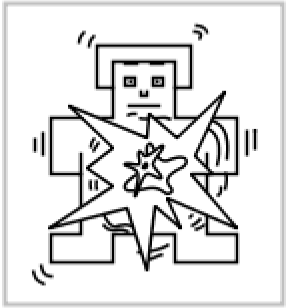
B



C



D



E

variable

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

assignment operator

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

variable types such as int, double

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

while loop

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

for loop

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

if statement

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

pseudocode

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

Boolean expressions, &&, II

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

subroutines / functions / methods

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

Java classes

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

debugger

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

references (boxes and arrows)

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

```
int [] array = new int[5] ;
```

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

Scanner

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

good coding style

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

Exception

A. never heard of this concept

B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

constructor

A. never heard of this concept

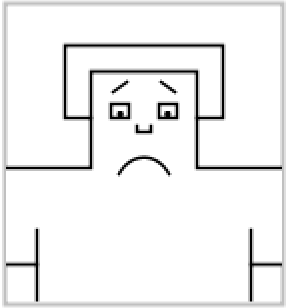
B. have heard, but don't know concept

C. I have some idea about the concept but do not know when or how to use

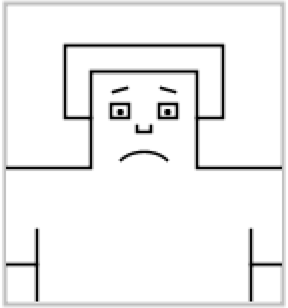
D. I have a clear understanding of this concept, but have not applied it in practice

E. I can explain what the concept is and have applied it in practice

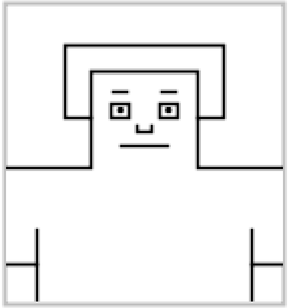
“Algorithms & Data Structures”:



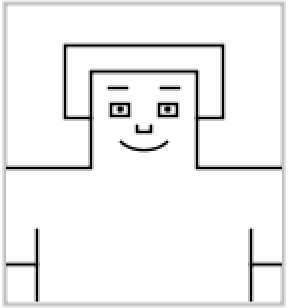
A



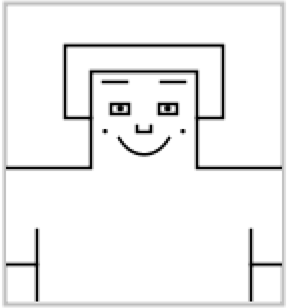
B



C

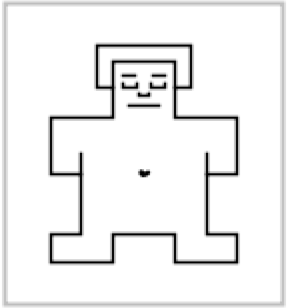


D

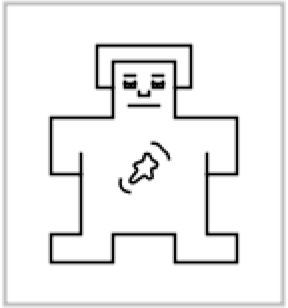


E

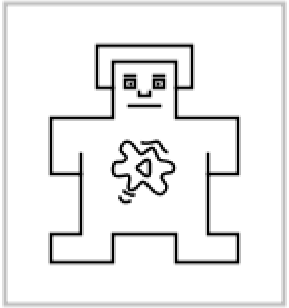
“Algorithms & Data Structures”:



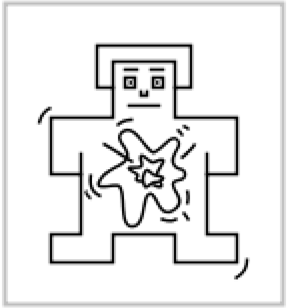
A



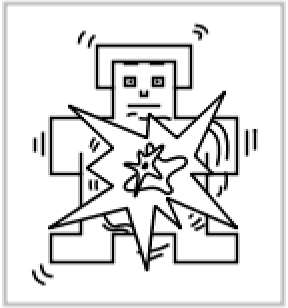
B



C



D



E

project 1

available on Learn now

weekend homework: read over entire project

discuss on Monday

next class (and most of next week):

- abstraction & classes
- discuss what data structures you *already* know