COSC 111:
Computer Programming I

Dr. Bowen Hui

University of British Columbia
Okanagan

Course overview

- Course outline

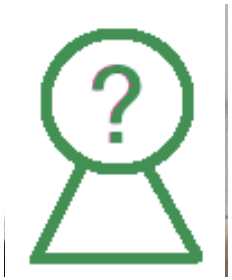
Course overview

- Course outline
- Today:
  - Course logistics, policies, expectations
  - Concepts and learning outcomes

Course overview

- Course outline
- Today:
  - Course logistics, policies, expectations
  - Concepts and learning outcomes
- Name plates (part of 5%)
  - Introductions (2 minutes)
    - Meet 3 other people
    - What is the one thing they all have in common?

Who we are



Bowen

Duncan

Ethan

Raffi

Matt

Rodney

Jessica

Anna

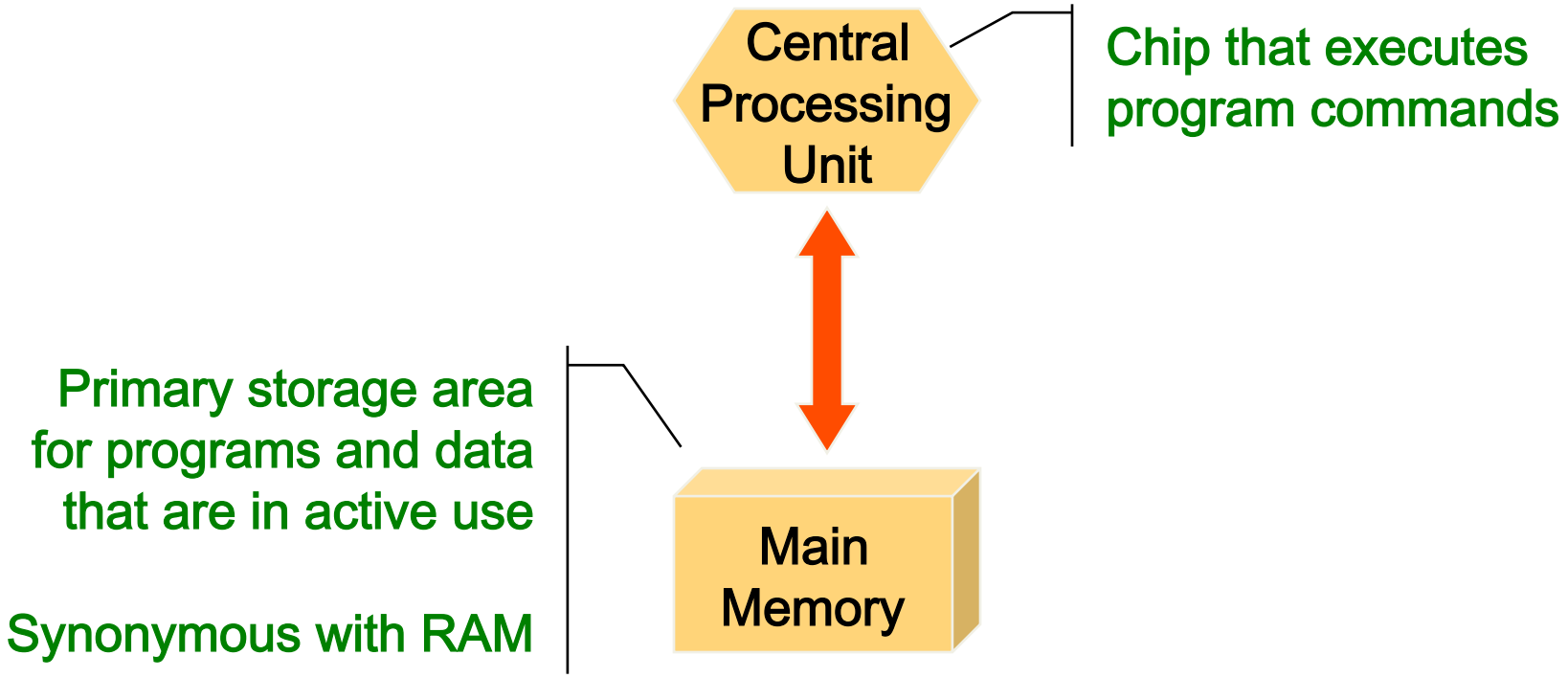
Peter

- Commonality: we're all excited to try out the new labs!

Hardware vs. Software

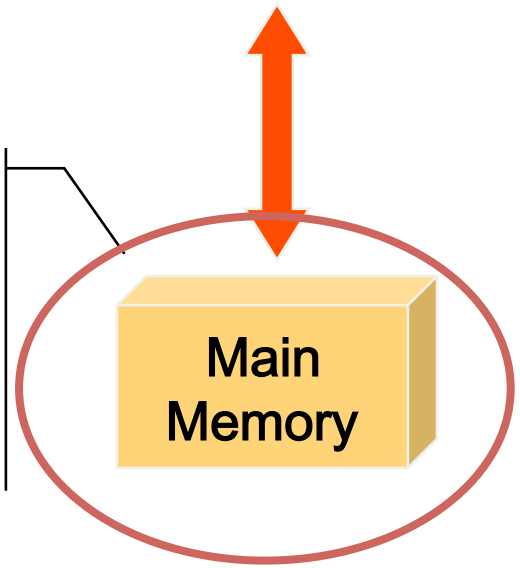
- Hardware
  - Physical, tangible parts of a computer
  - Keyboard, monitor, disks, wires, chips, etc.
- Software
  - Programs and data
  - Program = series of instructions (commands you issue to the computer)
- A computer requires both

CPU and Main Memory



CPU and Main Memory

Chip that executes program commands

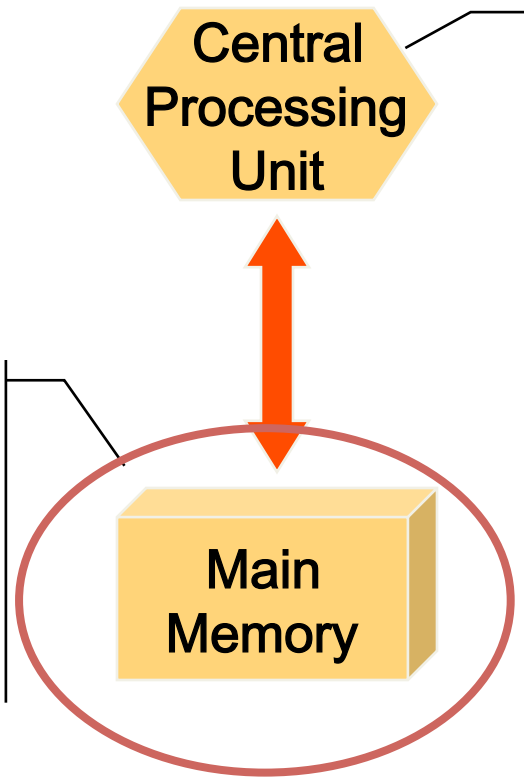


Primary storage area for programs and data that are in active use

Synonymous with RAM

What is this for?

CPU and Main Memory



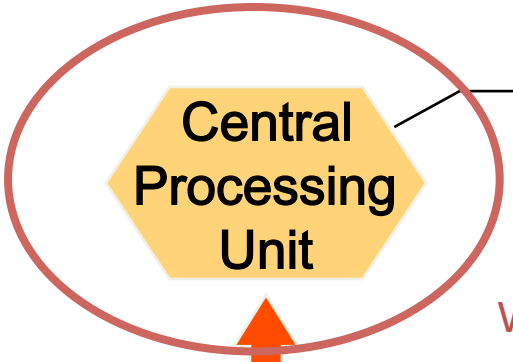
Chip that executes program commands

Primary storage area for programs and data that are in active use

Synonymous with RAM

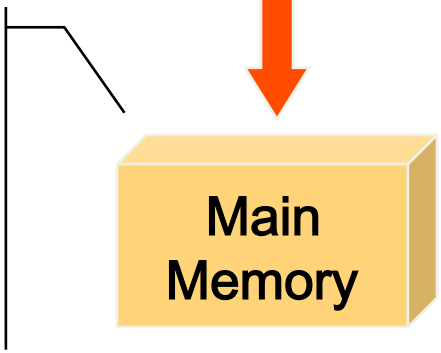
What is this for?
Think about a conversation when you make a lot of references to other people…

CPU and Main Memory



Chip that executes program commands

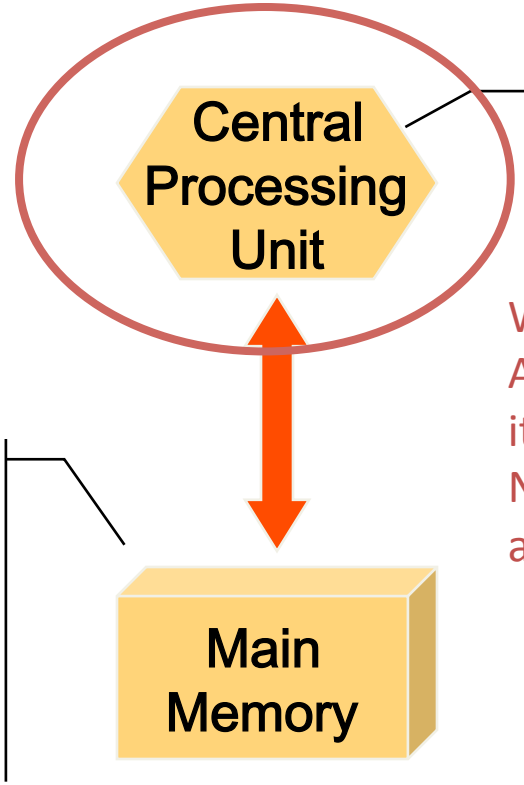
What is this for?



Primary storage area for programs and data that are in active use

Synonymous with RAM

CPU and Main Memory



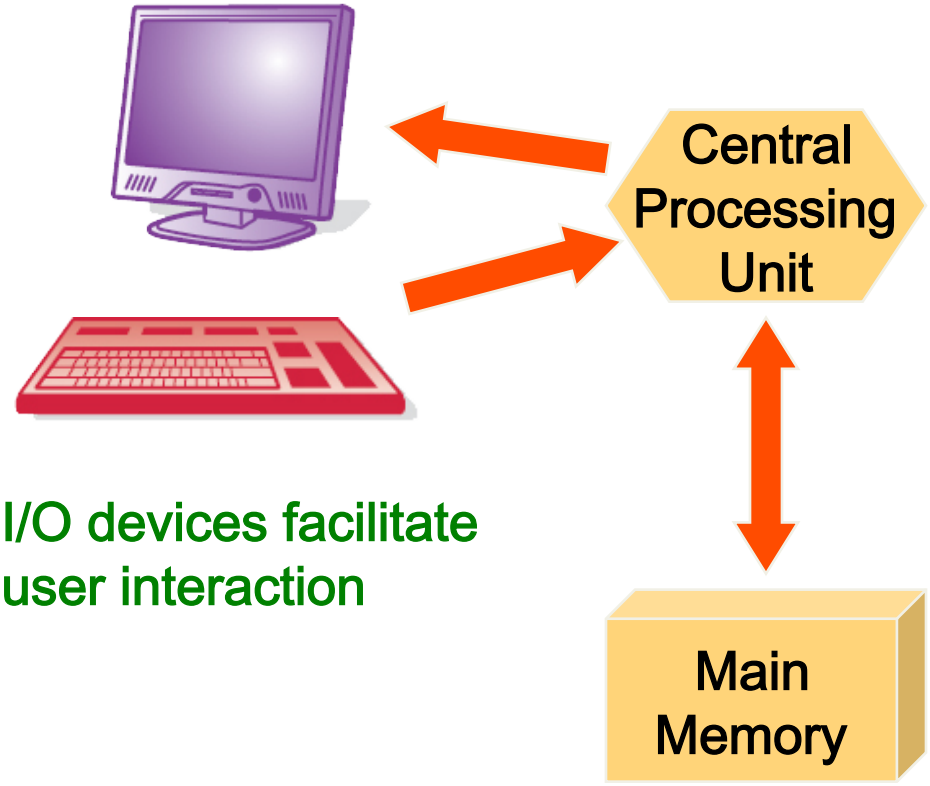
Chip that executes program commands

What is this for?
A computer is a “slave”, you give it tasks to do by issuing commands
Not all computers are equal, some are faster than others

Primary storage area for programs and data that are in active use

Synonymous with RAM

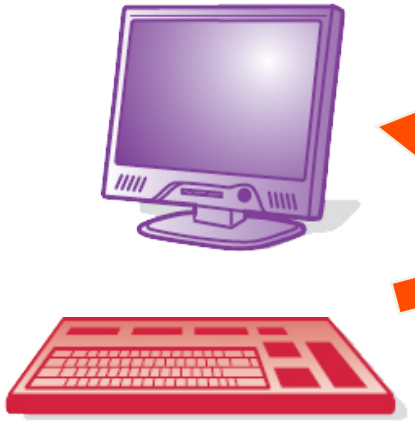
Input/Output Devices



I/O devices facilitate user interaction

Monitor screen
Keyboard
Mouse
Touch screen

Secondary Memory Devices



Central Processing Unit

Information is moved between main and secondary memory as needed

Hard Disk

Main Memory

Secondary memory devices provide long-term storage

USB Flash Drive

What is this course about?

- How to make computer programs
  - Process of building software
  - Relationship to hardware and input devices
  - Translating description to requirements and design (all in English)
  - Working with a programming language called Java
  - Process of testing your own software
- No programming background assumed

Example Programs

- Calculator-type applications
  - E.g. Processing/automation

Example Programs

- Calculator-type applications
  - E.g. Processing/automation
- Simple software with user input/output
  - E.g. Guess a number game

Example Programs

- Calculator-type applications
  - E.g. Processing/automation
- Simple software with user input/output
  - E.g. Guess a number game
- Applications with decision flow
  - E.g. Choose your own adventure game

Example Programs

- Calculator-type applications
  - E.g. Processing/automation
- Simple software with user input/output
  - E.g. Guess a number game
- Applications with decision flow
  - E.g. Choose your own adventure game
- More complex processes
  - E.g. Library loans

Demos

- Porter Stemmer
- Robot

```
map:
- kitchen (K) - tea
- door (D) - newspaper
- office (O) - where the master is
- library (L)
- hallway (H)
O - K -L
    |              NORTH is up
    H - D
```

- Trivia
- FeedMe

Relation to Mathematics

- Attention to details
  - E.g. x is not same as X

Relation to Mathematics

- Attention to details
  - E.g. x is not same as X
- Use of variables and functions
  - E.g. use variables to store values
  - E.g. use methods to carry out sequence of steps

Relation to Mathematics

- Attention to details
  - E.g. x is not same as X
- Use of variables and functions
  - E.g. use variables to store values
  - E.g. use methods to carry out sequence of steps
- Logical and conditional statements
  - E.g. if *condition* then *consequences*

Relation to Mathematics

- Attention to details
  - E.g. x is not same as X
- Use of variables and functions
  - E.g. use variables to store values
  - E.g. use methods to carry out sequence of steps
- Logical and conditional statements
  - E.g. if *condition* then *consequences*
- Algorithms to solutions
  - E.g. prove your solution works

Relation to Mathematics

- Attention to details
  - E.g. x is not same as X
- Use of variables and functions
  - E.g. use variables to store values
  - E.g. use methods to carry out sequence of steps
- Logical and conditional statements
  - E.g. if *condition* then *consequences*
- Algorithms to solutions
  - E.g. prove your solution works
- Advanced applications (not in this course)
  - E.g. calculus, probability, geometry, graph theory, etc.

Learning Outcomes

- Students will be able to …
  - Develop an appreciation for the complexity in creating computer software
  - Build simple programs on your own from scratch
  - Contribute to existing programs given to you
  - Find relevant resources to help trouble shoot programming problems
  - Work with others to come up with solutions

Why – appreciate complex software?

Why – appreciate complex software?

- Software development process
- Related technical jobs
  - Project managers
  - System analyst
  - Marketing
- Hire/Work with programmers
- Explore job opportunities in booming industry

Why – build your own?

Why – build your own?

- Famous proverb
  - I see and I forget. I hear and I remember. I do and I understand.
- Experiential learning (learn by doing)
- Basic skills in the computing industry
  - Helps with related fields (math, biology, stats, …)
  - Helps with related jobs
- Improve problem solving skills
- Sense of accomplishment

Why – add to others?

Why – add to others?

- Easier to start
- Don’t have to re-invent the wheel
- Working together
  (even without a structured team)
- Get more done
- Sense of community
- Higher possibility to be used by other people
- Potential to contribute to open source projects worldwide
  (volunteer your own software)

Why – resource identification?

Why – resource identification?

- Mimic real world (outside of school) environment
  - Find relevant resources
  - Evaluate reliability of resources
  - Integrate others' solutions into your own (in/outside of your team)
  - Properly credit others' work
- Applicable to every industry

Why – work with others?

Why – work with others?

- It’s inevitable!
- It’s what employers want
- More productive than working solo
- More social/fun/creative
- Communication skills is heavily valued in every industry

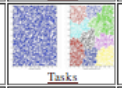
Evaluation Criteria

- 20% Weekly labs (10 total)
- 15% Assignments (3 total + 1 bonus)
- 5% In-class participation
- Exams:
  - 10% Midterm 1
  - 20% Midterm 2
  - 30% Final exam (cumulative)

Evaluation Criteria

- 20% Weekly labs (10 total)
- 15% Assignments (3 total + 1 bonus)
- 5% In-class participation
  - Alternative for those with programming experience – see me after class
- Exams:
  - 10% Midterm 1
  - 20% Midterm 2
  - 30% Final exam (cumulative)

Tentative Schedule

Week	Topics	Research Showcase	Readings	Assignments and Tests	Labs
1 (09/02)	• Tues: Class cancelled (Orientation day) • Thurs: Course overview, Software programs (slides)		• N/A • Ch 1.1-1.2		Labs cancelled
2 (09/09)	• Tues: Software Development Process: Requirements, Design, Input-Process-Output (IPO), Java Program Structure (slides) • Thurs: Requirements, Design, Attributes, Methods	 Tasks	• Ch 7.1-7.2, 7.7 (up to pg. 333), Ch 1.6, 1.5, 1.4, 4.1 • N/A	Thurs: A0 due (bonus)	Lab 1
3 (09/16)	• Tues: Classes, Objects, Variables, Statements • Thurs: Java API, Method Stubbing in Design and Implementation (ESL.java, TestESL.java)	 Tasks	• Ch 3.1, 4.2, 4.4, 4.5, 2.1-2.4 • Ch 3.2		Lab 2
4 (09/23)	• Tues: Class relationships, Visibility Modifiers, other Java classes • Thurs: Multiple Classes and Objects, Iterative development	 Task 1, Task 2, Task 3	• Ch 7.4, 4.3, 2.6, 3.4, 3.5 • N/A	Thurs: A1 due	Lab 3
5 (09/30)	• Tues: Pass by Value, Pass by Reference • Thurs: Class Interactions and Dependencies (Topic.java, TestTopic.java, Penpal.java, TestPostcard.java)	 Tasks	• Ch 7.7 • N/A		Lab 4
6 (10/07)	• Tues: Review • Thurs: Midterm exam		• N/A • N/A	Thurs: Midterm 1	Lab 5
7 (10/14)	• Tues: Modeling decisions: Conditional Logic • Thurs: Software Development Process: Testing (SentimentAnalysis.java)	 Tasks	• Ch 5.1-5.3, 7.9 • N/A		Lab 6
8 (10/21)	• Tues: Modeling repetitions: Loops • Thurs: Loop applications (Cluster.java)	 Tasks	• Ch 5.4, 6.3-6.4 (skip for-each loops) • N/A	Thurs: A2 due	Lab 7
9 (10/28)	• Tues: Arrays • Thurs: Multidimensional Arrays (StateTransition.java, TestTransitions.java)	 Tasks	• Ch 8.1-8.3, 8.6 • N/A		Lab 8
10 (11/04)	• Tues: Review • Thurs: Midterm exam		• N/A • N/A	Thurs: Midterm 2	Lab 9
11 (11/11)	• Tues: Graphical User Interface (GUI) • Thurs: Graphical User Interface (GUI)		• Ch 2.7, 3.9-3.11, 4.6-4.9, 6.5 • Ch 5.7, 8.8-8.9, 9.8		Lab 10
12 (11/18)	• Tues: Class cancelled (Midterm break) • Thurs: Static members and Overloading		• N/A • Ch 7.3, 7.8	Thurs: A3 due	Labs cancelled (Tues labs only) TA Office Hours (Wed-Fri labs)
13 (11/25)	• Tues: Preview of COSC 121 • Thurs: Games Show & Tell, Review		• N/A • N/A		TA Office Hours (Tues labs only) Labs cancelled (Wed-Fri labs)

A lot of work!

- Keep on top of the readings and lectures
- Get help AS SOON AS you sense you're falling behind
- Participate actively and interactively
- Practice programming regularly

Late Policy

- Labs and assignments
  - Due at start of lab/class
    - E.g., 10:59AM not 11:00AM
    - Very stringent
  - No lates accepted without a medical note

Missed Exams

- Missed midterms:
  - Receive a mark of 0% without a valid medical note
  - With a valid medical note: missed exam portion will be combined with other exams (all exams taken will still be worth 60% of the course grade)
- Missed final:
  - Receive a mark of 0% without a valid medical note
  - With a valid medical note accepted by the Dean's Office, a make-up exam will be scheduled

Passing Criteria

- Students must achieve a passing grade in the final exam in order to pass the course
- Students must achieve a passing grade in the lab component in order to pass the course
- Otherwise:
  - Student will receive a max of 45% as the final grade

Expectations in lectures

- Format:
  - Lectures (interaction!!!)
  - Slides & board notes
  - Group activities with TA support
  - Weekly “showcase” of cool applications
- You need to:

Expectations in lectures

- Format:
  - Lectures (interaction!!!)
  - Slides & board notes
  - Group activities with TA support
  - Weekly “showcase” of cool applications
- You need to:
  - Take notes (this is not a 39-hour movie)
  - Participate in group activities
  - Listen actively
  - Ask questions
  - Provide answers

Expectations in labs

- Pre-lab exercises
  - Handwritten exercises
- In-lab activities
  - Programming exercises
  - Lead to your own games by end of semester!
- You need to:

Expectations in labs

- Pre-lab exercises
  - Handwritten exercises
- In-lab activities
  - Programming exercises
  - Lead to your own games by end of semester!
- You need to:
  - Attend every lab on time
  - Submit pre-lab exercises at beginning of each lab
  - Pair program with someone
  - Complete lab activities (one week to submit)

Expectations on assignments

- Mix of written and programming questions
- Work with one other person of your choice or work solo
- You need to:

Expectations on assignments

- Mix of written and programming questions
- Work with one other person of your choice or work solo
- You need to:
  - Work on assignment questions honestly
  - Follow instructions on assignments to get points
  - Submit on time (very stringent)

Expectations on exams

- All individual, handwritten (even programming)
- One 8”x11” cheatsheet allowed
  - Helps organize notes into one spot
  - Process of writing notes reinforces concepts
  - Organization makes relationships more clear
  - Provide (helpful) starting point of reference for your answers
- You need to:

Expectations on exams

- All individual, handwritten (even programming)
- One 8”x11” cheatsheet allowed
  - Helps organize notes into one spot
  - Process of writing notes reinforces concepts
  - Organization makes relationships more clear
  - Provide (helpful) starting point of reference for your answers
- You need to:
  - Study
  - Understand examples from classes, labs, assignments
  - Sleep well the night before the exam

Tentative Schedule

- What is computer software?
- From English to computer language
- Object oriented programming (OOP)
- Control flow
- Data structures
- Graphical user interfaces (GUI)
- Special topics

Tentative Schedule

- What is computer software?
  - Relation to hardware and input devices
  - Examples of software applications
- From English to computer language
- Object oriented programming (OOP)
- Control flow
- Data structures
- Graphical user interfaces (GUI)
- Special topics

Tentative Schedule

- What is computer software?
- From English to computer language
  - Software development process
  - Generating requirements & design
- Object oriented programming (OOP)
- Control flow
- Data structures
- Graphical user interfaces (GUI)
- Special topics

Tentative Schedule

- What is computer software?
- From English to computer language
- Object oriented programming (OOP)
  - Classes, objects, attributes, methods
  - Abstraction
  - Encapsulation
  - Class interactions
- Control flow
- Data structures
- Graphical user interfaces (GUI)
- Special topics

Tentative Schedule

- What is computer software?
- From English to computer language
- Object oriented programming (OOP)
- Control flow
  - Program control flow
  - Logic statements
  - Conditional statements
- Data structures
- Graphical user interfaces (GUI)
- Special topics

Tentative Schedule

- What is computer software?
- From English to computer language
- Object oriented programming (OOP)
- Control flow
- Data structures
  - Arrays (single and multidimensional)
  - Java collections
- Graphical user interfaces (GUI)
- Special topics

Tentative Schedule

- What is computer software?
- From English to computer language
- Object oriented programming (OOP)
- Control flow
- Data structures
- Graphical user interfaces (GUI)
  - Java Swing, JFrame, JPanel, text, images, shapes, etc.
  - Event handling
  - 2D animation, paint, collision
- Special topics

Tentative Schedule

- What is computer software?
- From English to computer language
- Object oriented programming (OOP)
- Control flow
- Data structures
- Graphical user interfaces (GUI)
- Special topics
  - Scoping
  - Overloading

Resources

- Me
- TAs
- Course website
- Lab manual
- Course discussion forum on Blackboard Connect
- Textbook
- Math and Science Tutoring Centre in UNC 201
- Others:
  - You will find many others… we don't necessarily endorse them, but we can't prevent you from using them
  - Use with caution!
  - Don't plagiarize!

Readings

- Today:
  - Ch 1.1 – 1.2

- Next class:
  - Ch 7.1-7.2, 7.7 (up to pg. 338) software process
  - Ch 1.4-1.6 Java programming and environment
  - Ch 4.1 classes and objects

Lab 1 & A0

- Labs start next week
  - First one is before Tuesday's lecture
  - Complete pre-lab before arriving to lab
  - Due: following lab
- Bonus assignment to help familiarize you with
  - The course website
  - The technology for submitting assignments
  - Our submission process
  - Due: next Thursday 11:59pm