Introduction to Computer and Programming

Chapter 0: Course information

Manuel

Fall 2018

Outline

1. Logistics
2. Evaluations
3. Resources

Who?

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
  - Jiayi (jane_chen@sjtu.edu.cn)
  - Zihao (shenzihao@sjtu.edu.cn)
  - Zhi (linzhilynn@gmail.com)
  - Shuhan (harrywsh@sjtu.edu.cn)

Who?

Teaching team:

- Instructor: Manuel (charlem@sjtu.edu.cn)
- Teaching assistants:
  - Jiayi (jane_chen@sjtu.edu.cn)
  - Zihao (shenzihao@sjtu.edu.cn)
  - Zhi (linzhilynn@gmail.com)
  - Shuhan (harrywsh@sjtu.edu.cn)

Important notes:

- When contacting a TA for an important matter such as updating a grade cc the message to the instructor
- Add the tag [vg101] to the email subject
  e.g. `Subject: [vg101] important issue`
- Do not send large files ($>$ 2 MB) by email, instead use SJTU jBox service

When?

Course organisation:

- Lectures:
  - Tuesday 10:00 – 11:40
  - Thursday 10:00 – 11:40
  - Friday 10:00 – 11:40 (even weeks)
- Lab sessions:
  - Monday 18:20 – 20:00
  - Wednesday 12:10 – 13:50
  - Thursday 18:20 – 20:00
- Recitation classes: will be announced on Canvas
- Office hours:
  - Tuesday 15:40 – 17:50
  - Other times available on appointment

What?

Main goals of this course:

- Design simple algorithms
- Understand the main concepts of programming
- Implement clearly stated algorithms in MATLAB/C/C++

What?

Main goals of this course:

- Design simple algorithms
- Understand the main concepts of programming
- Implement clearly stated algorithms in MATLAB/C/C++

Ultimate goal: understand programming and be able to quickly adjust to new languages/libraries

How?

Learning strategy:

- Course side:
  1. Getting familiar with programming (MATLAB)
  2. Understand deeper concepts (C)
  3. Bridge the gap between computer and human views (C++)

How?

Learning strategy:

- Course side:
  1. Getting familiar with programming (MATLAB)
  2. Understand deeper concepts (C)
  3. Bridge the gap between computer and human views (C++)
- Personal side:
  1. Read and write code
  2. Relate known strategies to new problems
  3. Perform extra research

Course outcomes

Detailed goals:

- Proficiency with data representation and naming
- Proficiency with data input and output
- Proficiency with programming with math and logical operators and functions
- Proficiency with designing, testing, and implementing functions and procedures
- Proficiency with control flow using selection and iteration
- Proficiency with use of pre-defined data structures
- Proficiency with primitive and complex data types
- Proficiency with visualization of data
- Proficiency with algorithm design for engineering analysis

Outline

1 Logistics

2 Evaluations

3 Resources

Assignments, labs, and projects

Assignments:

- Total: 8
- Content: basic algorithms, Matlab, C, C++
- Not graded, completed in groups or individually

Labs:

- Total: 8
- Content: guided sessions in Matlab, C, and C++

Projects:

- Total: 3
- Content: advanced problems in Matlab, C, and C++

Grading policy

Grade weighting:

- Matlab midterm: 20%
- C midterm: 20%
- C++ final: 20%
- Projects: 35%
- Labs: 5%

Grading policy

Grade weighting:

- Matlab midterm: 20%
- C midterm: 20%
- C++ final: 20%
- Projects: 35%
- Labs: 5%

Late submission: -10% per day, not accepted after 3 days

Final letter garde: curved to balance the three sections

Grading policy

Assignments:

- Not graded
- Each student must complete all the mandatory exercises
- Each student must review the code of each of his teammates
- A final improved version is submitted for each group
- Submissions should be successfully compiled or interpreted

Grading policy

Assignments:

- Not graded
- Each student must complete all the mandatory exercises
- Each student must review the code of each of his teammates
- A final improved version is submitted for each group
- Submissions should be successfully compiled or interpreted

Students not following these rules will receive a -5% deduction on their final course grade

Honor Code

General rules:

- Not allowed:
  - Reuse the code/work from other students or groups
  - Reuse the code/work from the internet
  - Give too many details on how to solve an exercise

Honor Code

General rules:

- Not allowed:
  - Reuse the code/work from other students or groups
  - Reuse the code/work from the internet
  - Give too many details on how to solve an exercise
- Allowed:
  - Share ideas and understandings on the course
  - Provide general directions on where or how to find information

Honor Code

Documents allowed during the exams:

- Part A: a mono or bilingual dictionary
- Part B:
  - The lecture slides with **notes on them** (paper or electronic)
  - A mono or bilingual dictionary

Group works:

- Every student in a group is responsible for his group submission
- If a student breaks the Honor Code, the whole group is sent to Honour Council

Special circumstances

Contact us as early as possible when:

- Facing special circumstances (e.g. full time work, illness...)
- Feeling late in the course
- Feeling to work hard without any result

Any late request will be rejected

Outline

1 Logistics

2 Evaluations

3 Resources

Canvas

On Canvas platform:

- Course materials and assignments
- Announcements and notifications
- Polls

References

Places to find information:

- MATLAB documentation
- *C for Engineers and Scientists* by Harry H. Cheng
- *Thinking in C++* by Bruce Eckel
- Search the web

References

Places to find information:

- MATLAB documentation
- *C for Engineers and Scientists* by Harry H. Cheng
- *Thinking in C++* by Bruce Eckel
- Search the web
- Do not use baidu

Key points

- Work regularly, do not wait the last minute/day
- Respect the Honor Code
- Go beyond what is taught
- Do not learn, understand
- Keep in touch with us
- Any advice/suggestions will be much appreciated

Thank you!