VG101 — Introduction to Computer and Programming

Course information

Manuel — UM-JI (Fall 2018)

1 Presentation

The focus of this course will be the understanding of computers, that is learning how they work such as being able to interact with them and have them fulfilling some given tasks through programming.

In order to fully benefit from this course, students are expected to only conduct and submit their own, personal work.

2 Teaching team

Details related to the instructor and Teaching Assistants (TAs) are summarized in the following table.

Instructor and TAs	Contact	Office hours	Location
Manuel	charlem@sjtu.edu.cn	Tuesday 15:40 – 17:50[1]	JI 437A
Jiayi	jane_chen@sjtu.edu.cn	Wednesday 20:30 – 22:30	TBA
Zihao	shenzihao@sjtu.edu.cn	Wednesday 18:00 – 20:00	TBA
Zhi	linzhilynn@gmail.com	Monday 16:00 – 18:00	TBA
Shuhan	harrywsh@sjtu.edu.cn	Thursday 20:20 – 22:20	TBA

3 Schedule

The Fall semester is 14 weeks long, including one week for the finals and one week for the national holidays.

Lectures:

- Tuesday 10:00 – 11:40
- Thursday 10:00 – 11:40
- Friday 10:00 – 11:40 (even weeks)

Labs:

- Monday 18:20 – 20:00
- Wednesday 12:10 – 13:50
- Thursday 18:20 – 20:00

Recitation classes: announced on Canvas

4 Syllabus

This course has been tailored with three main goals in mind:

- Understand how computers work
- Design clear algorithms
- Implement clearly stated algorithms in MATLAB, C and C++

[1] Appointments outside of the office hours can be taken by email.

Hence, at the end of this course, students should be provided with a solid basis for any further study of computer science; In particular they should have developed the programming skills necessary to engage in solving engineering problems.

These goals are fulfilled through the following course outcomes:

- 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

The detailed organisation of this course is given as follows:

Weeks 1–5	**Weeks 5–9**	**Weeks 9–13**
0. Course information	5. Introduction to C	9. Introduction to C++
1. Computer and programming	6. Basic C	10. Object oriented programming
2. MATLAB scripting	7. Arrays and pointers	11. Libraries and templates
3. Functions and recursion	8. Algorithms and efficiency	12. Beyond MATLAB, C, and C++
4. Advanced MATLAB		
First midterm	**Second midterm**	**Final exam**

Labs are to be taken very seriously. A lab session splits into two parts: during the first one, the TAs discuss various aspects of the previous assignment, while the second one consists in the "real time solution" of a given exercise. During this second part students are expected to think, work, and actively participate in the discussion. A student that is not working will be asked to **leave the room** and be counted as missing.

5 Grading policy

General rules

The final average will be composed of five "sub-grades", apportioned as follows:

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

Any late submission will result in a 10% deduction per day from the grade of the corresponding work. After three days no submission will be accepted.

For the final grade a curve will be applied such as to balance all the three sections of this course.

Extra rules

In order to improve the quality and clarity of the code the following rules will be enforced. Failing to comply with them will incur a -25% and -10% penalties in the projects and exams, respectively.

- Indent the source code;
- Use meaningful variable names;
- Use functions and sub-functions;
- Add simple comments to the source code;
- Close any file that was opened;
- Release any dynamically allocated memory;
- Use local variables only;
- Include a README file in **text format** in the submission;[2]
- Add a Bug section to the README file;
- Never use the `goto` instruction;
- Never use the `friend` keyword;
- Never use the `system()` function;
- Never write any function code in a header file;
- Never use `fflush(stdin)`;

Projects

In the project any submission that **cannot compile or be interpreted will not be graded**. Note that for the exams such submissions will be graded normally.

Exams

All the exams will feature a few questions, graded out of 4 marks, that any student who worked on the project should be able to properly answer. Those simple questions will be clearly highlighted on the exam paper. Since a student who worked on the projects should be able to easily answer them, anyone not answering them properly will have his corresponding grade adjusted with respect to the following rules:

- < 1 on the exam questions implies 0 to the project;
- <= 1.5 on the exam questions implies -50% on the project, no bonus;
- <= 2 on the exam questions implies -25% on the project, no bonus;
- <= 2.5 on the exam questions implies no project bonus;
- > 2.5 on the exam questions implies project grade unchanged;

Note that TAs are allowed to add their own rules such as to facilitate the grading process. Not following their policies might result in extra penalties.

Assignments

Assignments are to be completed in groups and are not graded. Groups can be freely formed and registered on Canvas (cf. announcements for more details on the process).

[2] In an exam a perfect code without README file will not receive any penalty.

Although assignments are in group and ungraded, **every single student** is expected to complete at least all the exercises marked as "mandatory". Each homework features two deadlines; on the first one all the students in a group should submit their personal code to be peer reviewed by the other group members. Two days later, the whole group should submit its final version which is expected to be an improved version based on the various approaches of the students and the provided comments.

For each assignment, students not complying with the following requirements will automatically receive a deduction **on their final course grade**:

- Students should individually complete all the mandatory tasks and attach their work to *a question addressed to their Piazza group at least two days* before the deadline of the group submission. Students late for the individual submission must still post a question: instead of containing their code it should feature (i) apologies to the reviewer, (ii) clear explanations on why the work is late, and (iii) a request for a new deadline *fitting the reviewer*. When the submission is ready the *initial question* should be edited and the individual code attached to it. The reviewer is allowed to reject the request and should set the deadline based on his/her own schedule. **(-0.5 mark)**
- A student should provide feedbacks to at least one teammate for each mandatory exercise. Each student should receive feedbacks on his individual submission. e.g. for a three students group: student1 $\rightarrow$ student2 $\rightarrow$ student3 $\rightarrow$ student1. **(-1 mark)**
- The final group submission, should successfully be compiled or interpreted. **(-1 mark)**
- Any group submission that is more than 24 hours late will be rejected. **(-2.5 marks)**

Extra notes on the assignments:

- Although it is allowed to meet in person to discuss the individual submissions and prepare for the group submission, a summary must be posted on Piazza: this should help students in their revisions and provide more precise feedbacks to the teaching team.
- When posting a question ensure it is clear and precise, while not encouraging Honour Code violations (e.g. "How to solve question 2?" or "How do you change the color of the traffic light?"). Include simple examples highlighting what the problem is.
- Posting questions including sample code is allowed at any stage of the assignment (even before the individual submission). Before posting such a question ensure it can *only be viewed inside the group*.
- At the end of the semester **up to 5 marks bonuses** will be awarded on the final grade to students with high quality contributions on Piazza.

6 Honor code

It is of a major importance for any submitted work to be the result of one own research and understanding. In particular it is not acceptable to reuse the work from another student, or downloaded from the internet. Students can however help each others in an up-building way by sharing ideas and understanding on the course.

If in any case code or details from a textbook or internet is reused, the source should be clearly stated such as not to induce any possible confusion.

According to JI Honor Code copying the work of others will result in **severe penalties**.

Exams

Each exam splits into two parts: during the first one **not document or computer is allowed**. For the second part, besides the necessary software (e.g. MATLAB, text editor, IDE...), only the following documents are allowed during the exams.

- The electronic version of the course slides with notes on them;
- The printed version of the course slides with notes on them;
- A mono or bilingual paper dictionary;
- On the final exam, the OpenGL classes and templates written can be reused; However this code must be personal and cannot be shared among several students;

Any document, material, or mean of information and communication not explicitly listed above is strictly prohibited. In particular a **non-exhaustive** list of forbidden materials is as follows.

- Assignments and labs (questions and answers);
- Notebooks or separate files containing notes;
- Calculators;

Note: during the exam only the allowed documents can be opened on the computer. A student found with other materials, such as projects or labs, opened on his computer will be directly sent to Honor Council and will not be allowed to complete the exam.

Group works

Students are fully responsible for the work they submit. In particular in case of plagiarism the whole group will be sent to Honor Council, not only the student who did plagiarise.

For **group assignments and projects** communication is encouraged and even mandatory between the various group members. However any communication between students of different groups will be considered a violation of the course policy and Honor Code. In particular even sharing hints on an exercise between groups is strictly forbidden.

Constructive discussions on the lecture slides is encouraged and recommended, as long as it does not lead to breaking to above rules.

Projects

Discussion on the **individual projects** is allowed as long as it remains at a "hinting level", i.e. only hints can be provided. In particular if two students share information to the extent of ending up with similar codes it means they have talked too much and as such have broken the course policy and the Honor Code.

Similarly any source of information must be cited in order to avoid any plagiarism issue. However citing the source does not mean the content can be copied and pasted, it should only be used as a hint or direction on how to tackle the problem.

7 General information

The following references and links can be used to find information relevant to the course.

- All the course related materials will be made available on Canvas.
- This course will be sub-divided into three main parts: (i) MATLAB, (ii) C and (iii) C++, each accounting for about a third of the total time; each one will be loosely based on the following materials: (i) MATLAB documentation, (ii) *C for Engineers and Scientists*, by Harry H. Cheng and (iii) *Thinking in C++*, by Bruce Eckel.
- The following websites can be used for further research on programming:
  - C++ reference: `http://en.cppreference.com/w/`
  - cpluscplus.com: `http://www.cplusplus.com/`
  - Stackexchange: `https://stackexchange.com/`
- **Never** use baidu as a search engine for questions related to programming.

To improve communication between the students and the teaching team please observe the following guidelines.

- Any student facing a special situation likely to impact his studies, such as serious illness or full time work, is expected to contact the teaching team as early as possible in order to discuss it and see if any solution can be found.
- When sending an email related to this course include the tag "[vg101]" in the subject (e.g. `Subject: [vg101] grade issue`).
- When contacting a TA for a grade issue or any other major problem send a carbon copy (`cc`) to the instructor. Not doing so might result in omissions, not up-to-date grades etc...If such problem occurs and I have no record of the issue, the request will be **automatically rejected**.
- Never attach a large file ($>$ 2 MB) to an email, use SJTU dropbox and only include the corresponding link in the email.
- Keep in touch with the teaching team, feedbacks and suggestions will be much appreciated.
- Neither the TAs nor the instructor will respond questions on the assignments if asked **less than 24 hours** before the deadline.
- Keep in touch with the teaching team, feedbacks and suggestions will be much appreciated.