CS1101S

Programming Methodology I

Theodore Leebrant

Studio 2 - Tutorial Group 8D

Admin Stuff

Safety Measures

- Masks on at all time
- Stay 1.5m apart
- Passing objects require disinfection (e.g. markers)
- Take attendance and picture of seating arrangement

Attendance Taking

Click here for photo submission

We will take photo when everyone is present.

Telegram Group

alternatively click here

About Me

- Theodore Leebrant
- Year 2 Computer Science & Mathematics + USP
- Living in Cinnamon College
- Telegram: @kagamination
- Email: theodoreleebrant@u.nus.edu

Introduce Yourself!

- Preferred name
- Major (Probably CS)
- Academic and Programming Background
- Where do you stay?
  Halls, RCs, Woodlands, Johor, Japan, Yishun, or elsewhere
- Why CS (or your major)?

About This Studio

- Safe space
  - Questions, mistakes, comments are welcome.
- Expectations
  - **Come to studio prepared for discussion!**
  - (Try to) finish the studio sheet.
  - If still cannot, at least read the questions.
- Workflow
  - We will start 5 minutes late or when 6/8 people comes, whichever earlier
  - 5 minutes for admin (attendance taking, path/mission/quest review)
  - 10 minutes for lecture and brief recap
  - The rest for studio sheet
  - I'll stay back for any questions if needed

Communication Channels

- LumiNUS announcements
  - Look out for this, profs will send general announcements
- Piazza
  - A forum for general questions. Use it to ask questions; if you have free time, you can go ahead and answer questions there.
- Emails
  - For more official stuff, e.g. submitting MCs if you are absent.
- Telegram (group)
  - For less official stuff, e.g. asking questions or sharing memes. You are welcome to PM me if you feel uncomfortable talking in the group.

Consultations

- PM me on Telegram (preferred) or drop an email
- Either group or 1-to-1 consultations are fine, keep it below 5 people.
- F2F (preferred) or online (through zoom/discord)
- Check for timing, at least 1 day ahead. Most free on Mondays.

Code of Honor

- My answers to homework, quizzes, exams, and contests will be my own work (except for assignments that explicitly permit collaboration).
- My answers to paths, missions, and quests are **written by myself.** I am allowed to discuss ideas and algorithms with other students, but will work out the actual solution myself.
- I will not make solutions to homework, quizzes or exams available to anyone else. This includes both solutions written by me, as well as any official solutions provided by the course staff.
- I will not engage in any other activities that will dishonestly improve my results or dishonestly improve/hurt the results of others.

How to do Missions and Quests

Stuck?

- Try to figure it out by yourself for another 30 minutes.
- Ask your fellow cadets
  1. Studio telegram group
  2. Piazza
  3. etc.
- Let me know, maybe the question is wrong.

Avenger Grading

- Missions and Quests
  - Marked by me
  - 18% of your grades from the XPs
  - Resubmissions allowed in the spirit of learning (ask me to unsubmit)
- Studio grading (5%)
  - Show up for attendance marks
  - Try to answer questions for particpation but no competition
  - Participation is not about looking smart or getting problems right
  - Make your mistakes here (instead of in the exams)
  - Help your fellow studio cadets if they struggle
- Mastery Checks ($3\% \times 2$)
  - Two checks on your learning progress during the semester
  - Form pairs and schedule appointment
  - MC1: Weeks 4-13, MC2: Weeks 9-13
  - 3 topics, 5 minutes per person per topic, Pass/Fail
  - Re-testing allowed.

Studio Grading

Out of 500XP per session:

- 200 for attendance (warm body)
- Reasonable contribution brings you up to 350
- Exceptional contribution brings you up to 400
- Replace me as TA to get 500

Remember: XP comes from many sources.

Recap

Expressions vs. Statements

- Expressions:
  - Something* that can be **evaluated**.
  - Examples: `5` is an expression, `3+4` is also an expression.
- Statements:
  - Programs that can be **executed**
  - Will trigger a computational process.
  - One statement is the expression statement, i.e. an expression followed by semicolon. For example: `5;` is a statement.

Related piazza post: click here

*This "something" is a valid combination of operands and operators.

Operators and Operands

Operators act on operands.

Examples:

- `-5`
- `!true`
- `2 + 3`
- `first >= second`
- `a && b`
- `x ? y : z` (conditional expression)

Constants and Functions

- Constant declaration
- Function declaration and application
- Function parameters and arguments

Example: https://share.sourceacademy.nus.edu.sg/constfunc

```
const x = 5;
function cube(num) {
    return num * num * num;
}
cube(3);
cube(x);
```

Environments and Conditional expressions

Environment: a mechanism that allows naming. Example:
https://share.sourceacademy.nus.edu.sg/conditionalexpressions

```
const age = 25;
age > 21 ? "can vote" : "cannot vote";
```

General form:

```
predicate ? consequent expression : alternate expression
```

Abstraction

Thinking at a higher level - you don't need to know how it works under the hood.
Classic example: car; has a lot of components, but you just use the car and drive.
CS1101S example: `mosaic` function re-used in mission 1 question 3 and 4. You don't need to care how it is implemented. Example.

Studio Sheet (and Photo Taking)

Additional Material

Mission Feedback

Most of you got the idea of what to do in missions, so keep up the good work.
In general, if I flag an issue, it's something that needs to be addressed.
Try to follow the Source styling guide available here for reference on how to make your code readable in terms of indentation and spacing. For parameter naming, try to have concise but meaningful names - there is a tradeoff between verbosity and readability.

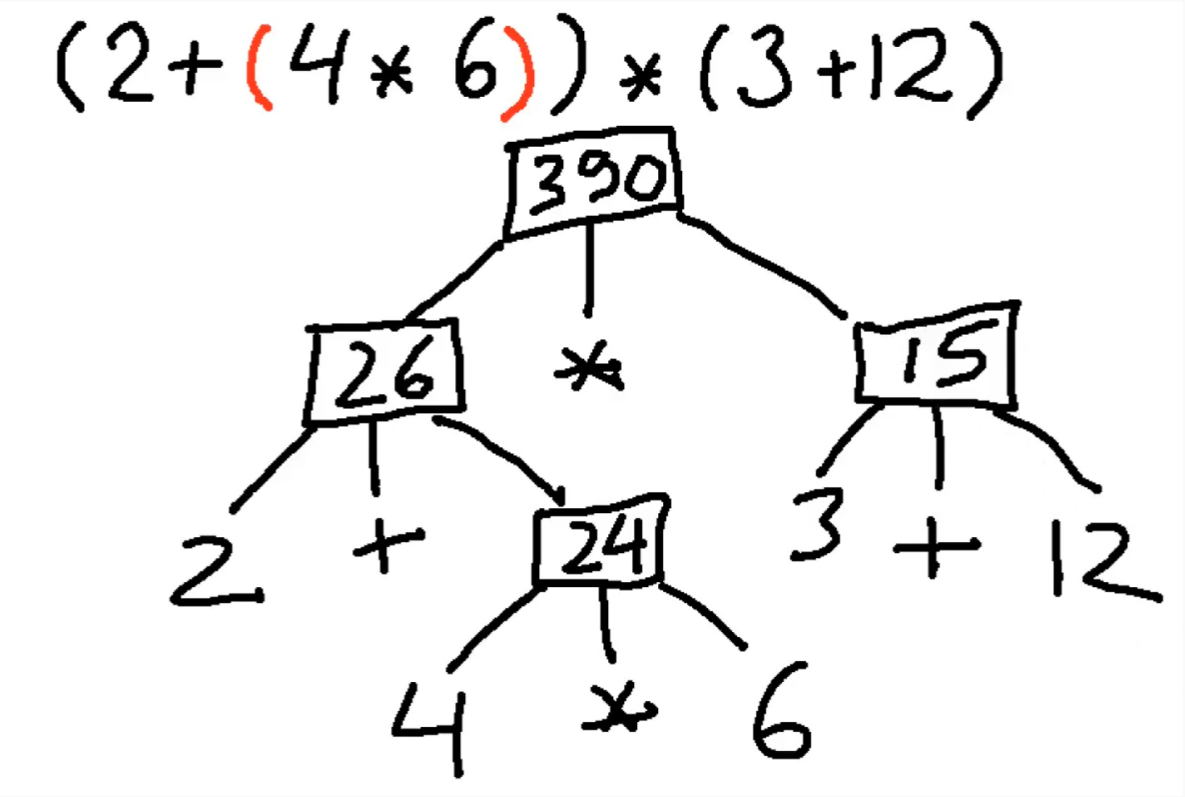
Mission Feedback - Commenting Style

Example

- Comments about the function goes above the function
  - You may use `\* ... *\` or the single line comment `\\`.
- Comments about single line goes either above the line or on the line.

Nugget N1: Evaluation of Expressions

`(2+(4*6))*(3+12)`



Click here for visualisation.