

CSC301 Phase 1 Project Planning - The Stack



Biographies

Joanne (Chia An) Weng:

I was born in Taipei, Taiwan and moved to Victoria, BC at the age of 13. I'm currently a fourth year computer science specialist student at University of Toronto. Throughout the years studying computer science, there are several interesting and useful courses that I had taken are: CSC309, CSC343, CSC454, CSC458, etc. I'm highly interested in working as a product/project manager (PM) after I graduate. In the summer of 2014, I worked at Synology's branch office in UK. During the internship, I was working as a tech support and project manager. One of the things that my manager assigned to me was in charged for the preparation and setup of the 2014 IFSec event in London.

Samson Chung:

My name is Samson Chung and I am a third year computer science student at UofT. I was born in Richmond Hill, just North of Toronto but currently live around campus for school. I am a member of New College's dragon boat team New Dragons and am currently their Finance Director. Being part of the team, I frequently go to the gym for fun and to run practices. I also like to play video games and hang out with friends. I enjoy computer science but am not sure what direction I want to go in within the field. I hope to learn more about software engineering through this class and help me decide my plans for the future!

Joseph Lee:

I was born in the large suburban area of Woodbridge, Ontario, which is just a few kilometres north of Toronto. I was raised in a Korean household along with two older brothers who study law and commerce, and both of them have gone to work overseas in different parts of Asia.

I attended St. Michael's College School, a Catholic high school in downtown Toronto, where I would spend most of his six years doing extracurricular activities and volunteering. After graduating, I went to pursue an undergraduate degree in Computer Science at the University of Toronto, where I am currently a 3rd year student. In the future I hope to start up my own company and hopefully enjoy what I do (in the computer science field).

In my spare time, I enjoy reading, playing sports, and travelling, if time permits it. But above all, I enjoy spending time with friends and family.

Calvin Cheng:

My name is Calvin Cheng. Born and raised in Toronto, I was interested in programming and computers at a very young age, and first started building websites when I was in grade 5. In high school I took a turn for the worst and went off path. Originally at UofT

studying Life Sciences for three and a half semesters - I dropped out for one semester, before coming back to specialize in computer science. My interest is in cloud computing, network security and mobile development. I currently am working on mobile app projects on the side, and I also drive for UberX on the side for some cash. My passion outside of school is Muay Thai, where I used to compete nationally with my team Southside Muay Thai located in Scarborough. I also like to read, code, do crazy things and go out and party on the weekends :). Upon graduating I hope to land a job at promising new startup; or be able to find some great minds to help put one together.

Andrew Trotter:

I was born in Toronto, Ontario and grew up in Mississauga. I attended high school in North York and am currently in my fourth year at University of Toronto. I am taking a double major in two very different subject areas: Computer Science and English Literature. I have worked in retail for 5 years and am looking forward to my liberation after I graduate. Aside from programming and literature, I have a passion for videogames and do small level design projects for FPS games in my spare time.

Alex Chang:

I was born in Taipei, Taiwan. At the age of 10, moved to British Columbia, Canada where I grew up and spent my elementary and highschool years. My hobbies include basketball, street dancing (popping), and video games. I am currently 3rd year in computer science, taking the specialist program in comp sci. I got my first taste of computer science and its applications whilst taking a course in my grade 12 year in high school. From there i gained an interest in computer science; how it worked and what made it worked. My first language that i programmed in is C. Right now, I am learning a variety of languages, such as C, Java, PHP, javascript, Python, SQL and much more. I am always willing and eager to learn. Upon graduating, I hope to land a job with an established company for experience purposes, from there i hope to join a small startup or create my own.

Jeremy Johnston:

I was born in Burlington, ON and lived there until moving to Toronto to come to the University of Toronto. I am currently a third year computer science specialist. I have mainly taken computer science classes throughout my time here including CSC300, CSC324, and CSC343. The electives I enjoyed were astronomy, economics, and environmental science. I hope to get an internship somewhere for the summer or year rather through PEY or some other method. The past few summer I've had a part time job working at a marina and restaurant. This is my first class where I've had a long term

project that is worked on for the entire semester so I hope to gain a lot of experience from it.

Choosing The Project

Our project, The Stack, is a web based educational tool designed for interactively teaching the basics of coding. Through its interactive tutorial style method in the form of a stacking game, The Stack, caters to computer-science students of beginner to novice levels as well as non-computer science students who wish to touch on the basics of coding. The game mainly teaches syntax and structure of coding through giving the player a working piece of code or algorithm that is placed out of order. The game works in level based puzzles where players can start in the beginner levels working their way up to subsequently harder levels. The main goal of each level is to reorganize the pieces, or Stacks, back into order given the input, to answer a specific output in order to clear the “level”. Further implementations we may add is structuring of algorithms, structuring of the stack and heap addresses, and even trees and graphs for teaching binary tree algorithms and graph algorithms at the advanced level.

We chose this project as we wanted to create a fun and enjoyable way of teaching computer science. Through gameplay, we have found a way to give learners an interactive element to learning computer science as oppose to the way of traditional learning through reading text.

We came to the decision to make “The Stack” as a group as we were originally discussing interactive ways to make learning computer science. We originally had the idea of making a game with different level designs to teach each topic, however we felt that the idea was too general. We got inspiration from candy crush for the level design. As we discussed the idea further, one of our group members brought up the idea of tetris. This gave us the idea of creating a stacking game, which in the end lead to the creation of “The Stack”

Personas

Persona #1

John Miller is a grade 12 student. He's highly interested in taking computer science courses in university. John has no experience in coding but wants to start learning some basic coding concepts in order to prepare him for university. John used some programming language tool books to learn the basic syntax of coding, but he's hoping to find a more interactive tool suitable for a beginner like himself. Many of the pre-existing coding practice websites are too hard for a beginner like John to learn from.

Persona #2

Jessie Lee is a 19 year-old statistics major at University of Toronto. One of the required courses for her program is CSC108/CSC148. Since she has no programming experience before, she decided to take CSC108. It has been three weeks since school started and Jessie is already struggling on understanding the course materials. She's hoping to find a beginner programming practice to help her improve coding skill that she has learned from CSC108 classes.

Persona #3

Daniel Chen is 21 year-old, international undergraduate computer science student. He is from Taiwan and coming to Canada is the first time has left Taiwan. He enjoys playing video games, including *League of Legends*. He also enjoys participating in sports such as swimming and basketball during his free time. He has novice level knowledge of computer science from his classes and has learned the basics of Python, C and Java. Daniel is now taking more difficult courses with languages that he is not as familiar with.

User Stories

- John Miller decides to search for a better website to practice computer science. John eventually finds The Stack. John decides to start a new beginner exercise in Python that he finds on The Stack.
- John has to create a for loop that keeps track of the sum of 100 randomly generated integers and write it to a text file. All the code is provided for John but the lines are in a mixed up order. John carefully deciphers the code and does not need to find external resources to solve the problem.

- John decides to learn more about algorithms and proofs and finds an exercise to re-stack a proof that is written and mixed up.
- Jessie finds her exercises in CSC108 to be really difficult because she is not that familiar with computer science. Jessie uses The Stack to find the relevant topics to the concepts she is learning in her CSC108 class.
- Jessie decides to do some of the problems she finds. With the code provided, using The Stack, she is able to keep track of her progress and what concepts she has practiced.
- Daniel is a gamer, so he would like an interactive way of learning instead of the traditional conventional way. Daniel does an activity on learning Python and immediately is interested in this game because of how it feels like it is a game with pop ups that give him “tips”.
- Daniel is a competitive gamer. Each exercise has a leaderboard and Daniel practices each exercise thoroughly so he can get a better score.
- Not only does Daniel want to do well at individual exercises, Daniel wants to climb the overall rankings for Python and eventually other languages too.

MVP

The Stack is a web-based learning game for people new to computer programming. The core feature is a drag-and-drop system in which the player can move around code snippets or “blocks”, solving technical challenges by putting the code in the right order and formation. Different constructs and control elements are colour-coded to reinforce the player’s memory, and multiple levels are available, each introducing new programming concepts such as loops and conditional statements. Each set of levels builds upon the previous set, but learners with prior experience can skip ahead to find a more challenging problem. Program output is displayed at the touch of a button, and the clever design of the stackable “blocks” guarantees that all code is complete and produces good output, so that the player can test their skills outside of the pre-made levels. By creating an account, the user also gets access to other great features like a level checklist that tracks their progress as they complete the levels. One of the technical challenges of a system that produces output on whatever the player enters is that an interpreter is needed to process their input. With such an interpreter, The Stack becomes a simple graphical programming language in itself.

No other project right now is as simple, yet effective as this. Many other products may need you to write code, but for those who have no idea can begin to visually see it laid out in front of them, and then move it around so they can learn. This idea does not stop at just code, it can and will expand to bigger ideas in computer science, such as data structures or sorting algorithms.

John is a user who has tested out The Stack, was ecstatic to find a website where he did not have to go to multiple sites to learn one concept. He was eager to learn Python and he experienced The Stack, he found out that it was not hard to learn through the program. Simply put, John quotes, “No other website satiates my need for a fun and educational computer science application.”

Release and Iteration Planning

The features we are planning on releasing in the first release are having the front page of our website, along with a functional prototype of the stacking game. We also hope to have a login feature, coupled with a login page. For the first iteration, we hope to have the main page of the website up and running with a possible server for the login details of the users. Since the game is the hardest part, we do not expect to have it done within the first week.

Postponed features include different puzzles for the game ranging in difficulties and languages, a progress tracking feature, social media links, and leaderboards. Our main reason for postponing these features is because they are not needed for our MVP. For our first release, we only really need the game and web pages. Being realistic in our time frame, having a basic working copy of the game along with the website should keep us plenty occupied. If we happen to get more time, having different puzzles would be the first of the postponed features that will be worked on.

CRC Cards

| <u>Game</u> | |
|------------------|---------------|
| Responsibilities | Collaborators |
| Levels | Level page |
| Game board | Profile |
| Progress | Home |
| Problem | Ranking |
| Logout | |

| <u>Home (Intro)</u> | |
|---------------------|---------------|
| Responsibilities | Collaborators |
| Login function | Level page |
| Register | Profile |
| About us | Q&A |
| Start game | Ranking |
| Logout | |

| <u>Login</u> | |
|------------------|---------------|
| Responsibilities | Collaborators |
| Username | Home |
| Password | Q&A |
| | Register |

| <u>Profile</u> | |
|------------------|---------------|
| Responsibilities | Collaborators |
| User info | Home |
| Progress | Q&A |
| Ranking | Ranking |
| Logout | Level page |

| <u>Ranking</u> | |
|---------------------------|---------------|
| Responsibilities | Collaborators |
| Show ranking of the level | Home |
| | Profile |
| | Q&A |
| | Level page |

| <u>Q&A</u> | |
|------------------|---------------|
| Responsibilities | Collaborators |
| Logout | Home |
| Show basic Q&A | Profile |
| | Level page |

| <u>Level Page</u> | |
|-------------------|---------------|
| Responsibilities | Collaborators |
| Show levels | Game |
| Select levels | Home |
| Logout | Profile |
| | Q&A |
| | Ranking |

| <u>Register</u> | |
|--|---------------|
| Responsibilities | Collaborators |
| User info: Username, password, email, name of user | Home |
| | Login |
| | Q&A |

User Story via CRC: (The bold is where they would go to)

1. John Miller decides to search for a better website to practice computer science. John eventually finds The Stack. John decides to start a new beginner exercise in Python that he finds on The Stack.

John would stumble onto the home page, then heads to the registration page to create and account and then start the game.

| <u>Home (Intro)</u> | |
|-----------------------------------|---------------|
| Responsibilities | Collaborators |
| Login function | Level page |
| Register | Profile |
| About us | Q&A |
| Start game (play the game) | Ranking |
| Logout | |

| <u>Home (Intro)</u> | |
|--|---------------|
| Responsibilities | Collaborators |
| Login function | Level page |
| Register (to create account to play game) | Profile |
| About us | Q&A |
| Start game | Ranking |
| Logout | |

| <u>Register</u> | |
|--|---|
| Responsibilities | Collaborators |
| User info: Username, password, email, name of user | Home (head to the home to play the game after accounts made) |
| | Login |
| | Q&A |

2. John decides to learn more about algorithms and proofs and finds an exercise to re-stack a proof that is written and mixed up.

John would head to the home page (assuming he is still logged in and on a different site) and then head to the game page, then to the levels page to change to algorithms and proofs.

| <u>Home (Intro)</u> | |
|------------------------------------|---------------|
| Responsibilities | Collaborators |
| Login function | Level page |
| Register | Profile |
| About us | Q&A |
| Start game (start the game) | Ranking |
| Logout | |

| <u>Level Page</u> | |
|---|---------------|
| Responsibilities | Collaborators |
| Show levels (show levels drops a table) | Game |
| Select levels (choose algorithms and proofs) | Home |
| | Profile |
| | Q&A |
| Logout | Ranking |

| <u>Game</u> | |
|-------------------|---------------|
| Responsibilities | Collaborators |
| Levels | Level page |
| Game board | Profile |
| Progress | Home |
| Problem | Ranking |
| Logout | |

3. John has to create a for loop that keeps track of the sum of 100 randomly generated integers and write it to a text file. All the code is provided for John but the lines are in a mixed up order. John carefully deciphers the code and does not need to find external resources to solve the problem.

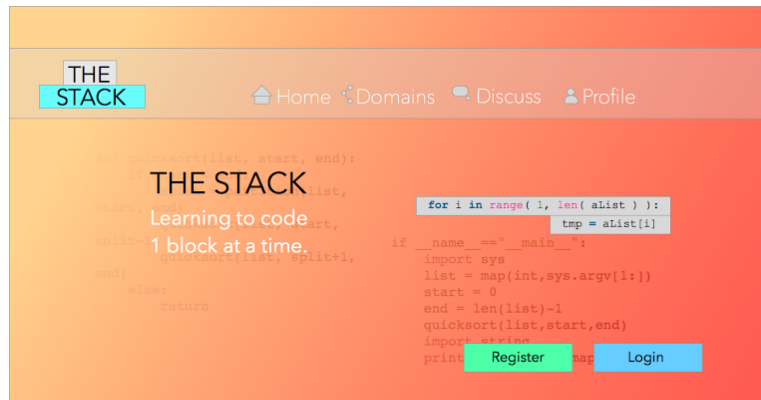
John would click start game (presumably on the home page) and head to the levels, change to “for loops” section and look at the problem of the game.

| <u>Home (Intro)</u> | |
|------------------------------------|---------------|
| Responsibilities | Collaborators |
| Login function | Level page |
| Register | Profile |
| About us | Q&A |
| Start game (start the game) | Ranking |
| Logout | |

| <u>Level Page</u> | |
|--|---------------|
| Responsibilities | Collaborators |
| Show levels (show levels drops a table) | Game |
| Select levels (choose for loops) | Home |
| | Profile |
| | Q&A |
| Logout | Ranking |

| <u>Game</u> | |
|-------------------|---------------|
| Responsibilities | Collaborators |
| Levels | Level page |
| Game board | Profile |
| Progress | Home |
| Problem | Ranking |
| Logout | |

UI Mockup Design



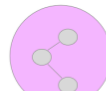
Learn

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non mauris nulla. Maecenas facilis leo vitae sapien ornare congue. Phasellus tempor dui in nisl rutrum euismod.



Interact

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non mauris nulla. Maecenas facilis leo vitae sapien ornare congue. Phasellus tempor dui in nisl rutrum euismod.



Solve

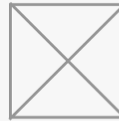
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non mauris nulla. Maecenas facilis leo vitae sapien ornare congue. Phasellus tempor dui in nisl rutrum euismod.



Code

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non mauris nulla. Maecenas facilis leo vitae sapien ornare congue. Phasellus tempor dui in nisl rutrum euismod.

About Us



THE STACK

[Home](#)
[Domains](#)
[Discuss](#)
[Profile](#)

user
[Logout](#)

```

import sys
alist = map(int,sys.argv[1:])
print alist

```

Algorithms -> Data Structures

Insertion Sort



```

for i in range(1, len( alist ) ):
    tmp = alist[i]

    k = i
    k -= 1

    while k > 0 and tmp < alist[k - 1]:

        alist[k] = alist[k - 1]

    def insertionsort( alist ):

        alist[k] = tmp

```

Run

Problem

Discussion

Sorting

One common task for computers is to sort data. For example, people might want to see all their files on a computer sorted by size. Since sorting is a simple problem with many different possible solutions, it is often used to introduce the study of algorithms.

Insertion Sort

These challenges will cover Insertion Sort, a simple and intuitive sorting algorithm. We will first start with an already sorted list.

Insert element into sorted list

Given a sorted list with an unsorted number V in the right-most cell, can you write some simple code to insert V into the array so it remains sorted?

Print the array every time a value is shifted in the array until the array is fully sorted. The goal of this challenge is to follow the correct order of insertion sort.

Guideline: You can copy the value of V to a variable, and consider its cell "empty". Since this leaves an extra cell empty on the right, you can shift everything over until V can be inserted. This will create a duplicate of each value, but when you reach the right spot, you can replace a value with V .

Input Format

There will be two lines of input:

- s - the size of the array
- ar - the sorted array of integers

Output Format

On each line, output the entire array every time an item is shifted in it.

Constraints

$1 \leq s \leq 1000$
 $-10000 \leq x \leq 10000, x \in ar$



Domains

| Algorithms | |
|-------------------------|-----------------|
| Sorting | |
| Data Structures | |
| Dynamic Programming | |
| Search | In Progress 2/3 |
| Quicksort | completed |
| Insertion Sort | completed |
| Bubble Sort | Do it! |
| Artificial Intelligence | |
| Machine Learning | |

