CS1101S: Programming Methodology

Martin Henz and Low Kok Lim

School of Computing

July 20, 2015
1 About CS1101S

2 Learning by playing, playing by learning
Basic principles

- CS approach: Programming Methodology (Sem 1) + Datastructures & Algorithms (Sem 2+3)
Basic principles

- CS approach: Programming Methodology (Sem 1) + Datastructures & Algorithms (Sem 2+3)
- At the moment in CS at SoC:
Basic principles

- CS approach: Programming Methodology (Sem 1) + Datastructures & Algorithms (Sem 2+3)
- At the moment in CS at SoC:
  - Programming Methodology (Sem 1): CS1010 or CS1101S
Basic principles

- CS approach: Programming Methodology (Sem 1) + Datastructures & Algorithms (Sem 2+3)
- At the moment in CS at SoC:
  - Programming Methodology (Sem 1): CS1010 or CS1101S
  - Datastructures & Algorithms (Sem 2+3): CS1020/CS2010 or CS2020
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: *Structure & Interpretation of Computer Programs*
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics, digital sound processing, and encryption
- In 2011, CS1101S turned into a game: assignments became missions, TAs became Avengers and students became apprentices in the magic arts of programming
- ...and in 2015, we will continue the story of CS1101S

Martin Henz and Low Kok Lim

CS1101S: Programming Methodology
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics, digital sound processing, and encryption
- In 2011, CS1101S turned into a game: assignments became missions, TAs became Avengers and students became apprentices in the magic arts of programming...
- ...and in 2015, we will continue the story of CS1101S
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics,
About CS1101S

Learning by playing, playing by learning

Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics,

... and in 2015, we will continue the story of CS1101S
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics, digital sound processing
### Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics, digital sound processing and encryption
Introduction to CS1101S

Learning by playing, playing by learning

About CS1101S

Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics, digital sound processing and encryption
- In 2011, CS1101S turned into a game: assignments became missions, TAs became Avengers and students became apprentices in the magic arts of *programming*...
Background of CS1101S

- Introduced in SoC (with variations) 16 years ago
- Follows didactic approach by Abelson and Sussman (MIT)
- Textbook: Structure & Interpretation of Computer Programs
- SoC adopted the material in 1997 and further developed it over the years
- Exciting practical elements such as robotics, computer graphics, digital sound processing and encryption
- In 2011, CS1101S turned into a game: assignments became missions, TAs became Avengers and students became apprentices in the magic arts of *programming*...
- ...and in 2015, we will continue the *story of CS1101S*
Central ideas

- Naming
- Functional abstraction
- Composition
- State
- Objects
Central ideas

- Naming
Central ideas

- Naming
- Functional abstraction
Central ideas

- Naming
- Functional abstraction
- Composition
Central ideas

- Naming
- Functional abstraction
- Composition
- State
Central ideas

- Naming
- Functional abstraction
- Composition
- State
- Objects
Clarification

No knowledge of any programming language is required!
Clarification

No knowledge of any programming language is required!

- For both CS1010 and CS1101S, you *do not need to know programming*
No knowledge of any programming language is required!

- For both CS1010 and CS1101S, you do not need to know programming.
- We actually prefer if students have no background in programming.
Clarification

No knowledge of any programming language is required!

- For both CS1010 and CS1101S, you do not need to know programming.
- We actually prefer if students have no background in programming.
- Recommendation: If you do have programming experience, come with an open mind, and not with the attitude: “I know this already.”
No knowledge of any programming language is required!

- For both CS1010 and CS1101S, you do not need to know programming.
- We actually prefer if students have no background in programming.
- Recommendation: If you do have programming experience, come with an open mind, and not with the attitude: “I know this already.” We promise that you will learn something new right in the first lecture already.
You will spend *a lot of time* with CS1101S.
You will spend *a lot of time* with CS1101S. You might as well have some *fun*. 
You will spend *a lot of time* with CS1101S. You might as well have some *fun*.

Completing your missions allows you to *level up*. 
You will spend *a lot of time* with CS1101S. You might as well have some *fun*.
Completing your missions allows you to *level up*.
Missions, trainings, side quests and contests are woven into a semester-long game.
You will spend *a lot of time* with CS1101S. You might as well have some *fun*.

Completing your missions allows you to *level up*.

Missions, trainings, side quests and contests are woven into a semester-long game.

Don’t worry, everyone will *win* in this game:
You will spend *a lot of time* with CS1101S. You might as well have some *fun*.

Completing your missions allows you to *level up*.

Missions, trainings, side quests and contests are woven into a semester-long game.

Don’t worry, everyone will *win* in this game: concepts, skills, experience,
You will spend *a lot of time* with CS1101S. You might as well have some *fun*.

Completing your missions allows you to *level up*.

Missions, trainings, side quests and contests are woven into a semester-long game.

Don’t worry, everyone will *win* in this game: concepts, skills, experience, passion for computing,
You will spend *a lot of time* with CS1101S. You might as well have some *fun*.

Completing your missions allows you to *level up*.

Missions, trainings, side quests and contests are woven into a semester-long game.

Don’t worry, everyone will *win* in this game: concepts, skills, experience, passion for computing, new friends.
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT. CS1101S touches upon many areas of computer science: computer graphics, digital sound processing, computer security, robotics, computer games, algorithms, and programming languages.

Look around you. After CS1101S, we hope you have an idea what computer science is all about and can look into specializations, minors and special programmes.
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT.
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science:
- computer graphics,
- digital sound processing,
- computer security,
- robotics,
- computer games,
- algorithms,
- programming languages.

Look around you.

After CS1101S, we hope you have an idea what computer science is all about and can look into specializations, minors, and special programmes.
What is computer science?
Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science:
- computer graphics,
- digital sound processing,
- computer security,
- robotics,
- computer games,
- algorithms,
- and programming languages.
CS1101S—A playful introduction to computer science

What is computer science?
Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science:
- computer graphics,
- digital sound processing,
- computer security,
- robotics,
- computer games,
- algorithms,
- programming languages.
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT

CS1101S touches upon many areas of computer science

computer graphics, digital sound processing, computer security,
About CS1101S
Learning by playing, playing by learning

CS1101S—A playful introduction to computer science

What is computer science?
Students usually have only a hazy idea of the field and equate computer science with IT

CS1101S touches upon many areas of computer science
computer graphics, digital sound processing, computer security, robotics,
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science:

- computer graphics
- digital sound processing
- computer security
- robotics
- computer games
- algorithms
- programming languages

Look around you. After CS1101S, we hope you have an idea what computer science is all about and can look into specializations, minors and special programmes.
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science:

- computer graphics
- digital sound processing
- computer security
- robotics
- computer games
- algorithms

Look around you. After CS1101S, we hope you have an idea what computer science is all about and can look into specializations, minors and special programmes.

Martin Henz and Low Kok Lim

CS1101S: Programming Methodology
What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science:

- computer graphics
- digital sound processing
- computer security
- robotics
- computer games
- algorithms
- programming languages

Look around you after CS1101S, we hope you have an idea what computer science is all about and can look into specializations, minors, and special programmes.
CS1101S—A playful introduction to computer science

What is computer science?

Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science.

computer graphics, digital sound processing, computer security, robotics, computer games, algorithms, and programming languages.

Look around you
What is computer science?
Students usually have only a hazy idea of the field and equate computer science with IT.

CS1101S touches upon many areas of computer science
- computer graphics
- digital sound processing
- computer security
- robotics
- computer games
- algorithms
- programming languages

Look around you
After CS1101S, we hope you have an idea what computer science is all about and can look into specializations, minors and special programmes.
The team

Avengers: your seniors, recruited to help you realize your potential

Lecturers: Dr Low Kok Lim and A/Prof Martin Henz

Creative and technical teams: Preparing your CS1101S experience under the direction of Ng Tse Pei and Evan Sebastian

Martin Henz and Low Kok Lim
Avengers: your seniors, recruited to help you realize your potential
The team

- Avengers: your seniors, recruited to help you realize your potential
- Lecturers: Dr Low Kok Lim and A/Prof Martin Henz
The team

- **Avengers**: your seniors, recruited to help you realize your potential
- **Lecturers**: Dr Low Kok Lim and A/Prof Martin Henz
- **Creative and technical teams**: Preparing *your* CS1101S experience under the direction of Ng Tse Pei and Evan Sebastian
To reflect the higher intensity and workload, CS1101S carries 5 MCs, whereas CS1010 carries 4 MCs.
Modular credits

- To reflect the higher intensity and workload, CS1101S carries 5 MCs, whereas CS1010 carries 4 MCs.
- For extra kicks, the game includes contests that allow you to get ahead of your peers in the leaderboard.
Modular credits

- To reflect the higher intensity and workload, CS1101S carries 5 MCs, whereas CS1010 carries 4 MCs.
- For extra kicks, the game includes contests that allow you to get ahead of your peers in the leaderboard.
- In addition, the more ambitious students can choose sidequests that complement the material and can be quite challenging.

If you manage to complete most of the sidequests, you get one additional MC through a special module called CS1010R. CS1010R is offered in Sem 2, but does not carry any workload. We will use the module to get to know future avengers and prepare improvements for 2016.
Modular credits

- To reflect the higher intensity and workload, CS1101S carries 5 MCs, whereas CS1010 carries 4 MCs.
- For extra kicks, the game includes contests that allow you to get ahead of your peers in the leaderboard.
- In addition, the more ambitious students can choose sidequests that complement the material and can be quite challenging. If you manage to complete most of the sidequests, you get one additional MC through a special module called CS1010R.
Modular credits

- To reflect the higher intensity and workload, CS1101S carries 5 MCs, whereas CS1010 carries 4 MCs.
- For extra kicks, the game includes contests that allow you to get ahead of your peers in the leaderboard.
- In addition, the more ambitious students can choose sidequests that complement the material and can be quite challenging. If you manage to complete most of the sidequests, you get one additional MC through a special module called CS1010R.
- CS1010R is offered in Sem 2, but does not carry any workload.
Modular credits

- To reflect the higher intensity and workload, CS1101S carries 5 MCs, whereas CS1010 carries 4 MCs.
- For extra kicks, the game includes contests that allow you to get ahead of your peers in the leaderboard.
- In addition, the more ambitious students can choose *sidequests* that complement the material and can be quite challenging. If you manage to complete most of the sidequests, you get one additional MC through a special module called CS1010R.
- CS1010R is offered in Sem 2, but does not carry any workload. We will use the module to get to know future avengers and prepare improvements for 2016.
So should you take CS1101S instead of CS1010?

- Do you want to go for a unique learning experience, using a semester-long game?
So should you take CS1101S instead of CS1010?

- Do you want to go for a unique learning experience, using a semester-long game?
- Do you not mind spending more time on this module than what’s reflected by the 5+1 modular credits?

Talk to the lecturers and avengers after this presentation
Read the FAQ “CS1010 and CS1101S”
Talk to seniors
Email to Dr Low and me (lowkl@comp.nus.edu.sg, henz@comp.nus.edu.sg)
So should you take CS1101S instead of CS1010?

- Do you want to go for a unique learning experience, using a semester-long game?
- Do you not mind spending more time on this module than what’s reflected by the 5+1 modular credits?
- Still undecided?
  - Talk to the lecturers and avengers after this presentation
  - Read the FAQ “CS1010 and CS1101S”
  - Talk to seniors
  - Email to Dr Low and me (lowkl@comp.nus.edu.sg, henz@comp.nus.edu.sg)
How to sign up?

Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S. Go to [https://register.comp.nus.edu.sg/UGOffice4/](https://register.comp.nus.edu.sg/UGOffice4/) and tick the box. The application opens on 20/7 at 5pm and closes on 23/7 at 6pm. Students who intend to join the Turing programme should indicate their interest in CS1101S. A quota currently limits enrolment in CS1101S. If the number of interested students exceeds the quota, it will be filled by simple random sampling. All interested students have equal chance as long as they tick the box before 23/7, 6pm. Students can check allocation results in CORS on 30/7.
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
- Go to https://register.comp.nus.edu.sg/UGOffice4/ and tick the box.
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
- Go to https://register.comp.nus.edu.sg/UGOffice4/ and tick the box.
- The application opens on 20/7 at 5pm and closes on 23/7 at 6pm.
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
- Go to https://register.comp.nus.edu.sg/UGOffice4/ and tick the box.
- The application opens on 20/7 at 5pm and closes on 23/7 at 6pm.
- Students who intend to join the Turing programme should indicate their interest in CS1101S.
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
- Go to https://register.comp.nus.edu.sg/UGOffice4/ and tick the box.
- The application opens on 20/7 at 5pm and closes on 23/7 at 6pm.
- Students who intend to join the Turing programme should indicate their interest in CS1101S.
- A quota currently limits enrolment in CS1101S. If the number of interested students exceeds the quota, it will be filled by *simple random sampling*. 
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
- Go to https://register.comp.nus.edu.sg/UGOffice4/ and tick the box.
- The application opens on 20/7 at 5pm and closes on 23/7 at 6pm.
- Students who intend to join the Turing programme should indicate their interest in CS1101S.
- A quota currently limits enrolment in CS1101S. If the number of interested students exceeds the quota, it will be filled by simple random sampling.
- All interested students have equal chance as long as they tick the box before 23/7, 6pm.
How to sign up?

- Students in the Double-Degree Programme in Computer Science and Mathematics are pre-allocated to CS1101S.
- Go to https://register.comp.nus.edu.sg/UGOffice4/ and tick the box.
- The application opens on 20/7 at 5pm and closes on 23/7 at 6pm.
- Students who intend to join the Turing programme should indicate their interest in CS1101S.
- A quota currently limits enrolment in CS1101S. If the number of interested students exceeds the quota, it will be filled by simple random sampling.
- All interested students have equal chance as long as they tick the box before 23/7, 6pm.
- Students can check allocation results in CORS on 30/7.
Common Misunderstanding: JavaScript is a kind-of Java

JavaScript and Java as different as English and Vietnamese
Common Misunderstanding: JavaScript is a kind-of Java

JavaScript and Java as different as English and Vietnamese

- Both are (programming/natural) languages.
Common Misunderstanding: JavaScript is a kind-of Java

JavaScript and Java as different as English and Vietnamese

- Both are (programming/natural) languages.
- They use a similar syntax (C-like/Latin alphabet).
Common Misunderstanding: JavaScript is a kind-of Java

JavaScript and Java as different as English and Vietnamese

- Both are (programming/natural) languages.
- They use a similar syntax (C-like/Latin alphabet).
- They have very different underlying structure.
In 1995, Brendan Eich started to work on an interpreted language for the Netscape browser, called Mocca, then LiveScript.

In December 1995, Sun Microsystems and Netscape announced the language under the name JavaScript.


Today, JavaScript is part of HTML5, the core technology for the World Wide Web.
Origins of JavaScript

From Mocha to HTML5

- In 1995, Brendan Eich started to work on an interpreted language for the Netscape browser, called Mocca, then LiveScript.
Origins of JavaScript

From Mocha to HTML5

1. In 1995, Brendan Eich started to work on an interpreted language for the Netscape browser, called Mocca, then LiveScript.

2. In December 1995, Sun Microsystems and Netscape announced the language under the name JavaScript.
Origins of JavaScript

From Mocha to HTML5

- In 1995, Brendan Eich started to work on an interpreted language for the Netscape browser, called Mocca, then LiveScript.
- In December 1995, Sun Microsystems and Netscape announced the language under the name JavaScript.
Origins of JavaScript

From Mocha to HTML5

- In 1995, Brendan Eich started to work on an interpreted language for the Netscape browser, called Mocca, then LiveScript.
- In December 1995, Sun Microsystems and Netscape announced the language under the name JavaScript.
- Standardized as ECMA-262, first edition in June 1997
- Today, JavaScript is part of HTML5, the core technology for the World Wide Web
JavaScript Today

- **Popularity**: Most widely "used" language on earth (in terms of individual "program" executions)
- **Simplicity**: Pure functional programming (dynamic types, Scheme-like)
- **Expressivity**: First-class functions, lexical scoping
- **Ugliness**: The full language is quite "idiosyncratic". Our approach: Focus on the good parts, a beautiful, clean, expressive sub-language that we've been calling *JediScript*.

Martin Henz and Low Kok Lim

CS1101S: Programming Methodology
JavaScript Today

Popularity

Most widely “used” language on earth (in terms of individual “program” executions)
JavaScript Today

**Popularity**
Most widely “used” language on earth (in terms of individual “program” executions)

**Simplicity**
Pure functional programming (dynamic types, Scheme-like)
# JavaScript Today

## Popularity

Most widely “used” language on earth (in terms of individual “program” executions)

## Simplicity

Pure functional programming (dynamic types, Scheme-like)

## Expressivity

First-class functions, lexical scoping
### JavaScript Today

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Popularity</strong></td>
<td>Most widely “used” language on earth (in terms of individual “program” executions)</td>
</tr>
<tr>
<td><strong>Simplicity</strong></td>
<td>Pure functional programming (dynamic types, Scheme-like)</td>
</tr>
<tr>
<td><strong>Expressivity</strong></td>
<td>First-class functions, lexical scoping</td>
</tr>
<tr>
<td><strong>Ugliness</strong></td>
<td>The full language is quite “idiosyncratic”. Our approach: Focus on the good parts, a beautiful, clean, expressive sub-language that we’ve been calling <em>JediScript.</em></td>
</tr>
</tbody>
</table>