Computer Science 101 S1

Principles of Programming

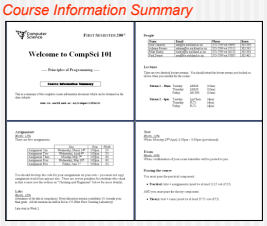
Lecture 1

Contents

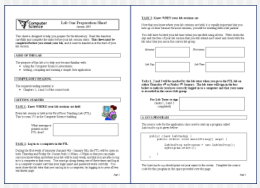
Introduction to the course
Algorithms
Programming steps

Handouts today

Course Information Summary



Lab One Preparation Sheet



People

Paul Denny (Course Supervisor)
Email: paul@cs.auckland.ac.nz
Phone: 373-7599 ext 87087
Room: 303.465

Mike Barley
Email: barley@cs.auckland.ac.nz
Phone: 373-7599 ext 86133
Room: 303.394

More People

Adriana Ferraro
Email: adriana@cs.auckland.ac.nz
Phone: 373-7599 ext 87113
Room: 303.592
Office hours: Wednesday 10:00 - 12:00,
Friday 11:15 - 1:00

Ann Cameron (Lab Supervisor)
Email: ann@cs.auckland.ac.nz
Phone: 373-7599 ext 84947
Room: 303.594

Things to do this week

- ✓ Read the Course Information Document online
- ✓ Get the CompSci 101 Coursebook
- ✓ Visit the First Floor Tutorial Lab (FTL)
- ✓ Complete your Lab One Preparation Sheet

7

Lectures

We will follow the course book quite closely

Any lecture slides will be available on the web before each lecture

8

Course Requirements

CompSci 101 Coursebook
$25 from the Student Resource Centre

Assessment

Labs	10%	PRACTICAL
Assignments	15%	
Test	15%	THEORY
Exam	60%	

To pass the course
you MUST pass the PRACTICAL (i.e. get 12.5 / 25)
you MUST pass the THEORY (i.e. get 37.5 / 75)

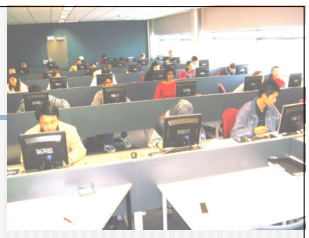
Labs

A lab preparation sheet (handed out in lectures and available on the web) must be completed **before** each lab starts.

You hand in the completed preparation sheet when you arrive at your lab.

At your lab time you will be given programming problems to solve within the 2 hours for your lab.

Labs



Assessment
Each lab worth 1%

Labs are held in the FTL lab

Labs start next week

Arrive on time

(This week, you need to visit the FTL lab to complete your lab preparation sheet)

Pair Programming

Pair-Programming
Two people sitting at the same computer and working together to produce a single program

Is Pair-Programming effective?
Produces higher quality code (less errors)
Can solve problems that can't be solved alone
More enjoyable than working alone

12

Pair Programming

Swap roles regularly

Joint work
The program is created and owned by both people
Both people must understand every line of code

Experienced programmers may be matched with novice programmers. There are advantages for both!

Advice
Be as helpful as you can!
Talk to each other

13

Assignments

Assignments provide experience with larger programs.

Solve the problem on your own – never share code for assignments.

Peer-review may be required for the assignments

Assessment
- Five assignments
- Worth 15% of your final mark
- Hand in your assignments using the web (using the Assignment Drop Box)

14

Assignments

Some examples of previous assignments by CompSci 101 students

15

Peer Review

Some of the CS101 assignments require a peer review using the Aropä system:

- Quick feedback
- Web-based
- Completed in a few days
- Anonymous

16

Peer Review

You are exposed to a variety of techniques and solutions

You can help others to learn

Be constructive with your comments!

17

Plagiarism

Any work that you take credit for, but which is done by someone else. This is treated very seriously in an academic environment.

Policy
- All assignments will be checked for copying
- *Everyone* involved is penalised
- Disciplinary action will be taken

Advice:
- Don't ever copy an assignment (or part of an assignment) from anyone
- Don't ever allow anyone to copy your assignment

18

Computing Resources

At Home (will discuss this on Thursday)

Java SE (Standard Edition) Development Kit (JDK)
- Free to download from the web

OR
- Computer Science Software CD (which can be bought for $10 from Student Resource Centre)

19

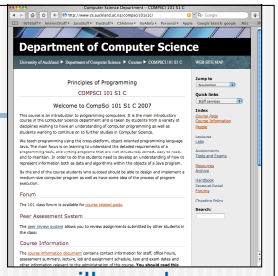
Computing Resources

Undergraduate Labs

FCL – first floor computer lab
OCL – first floor computer lab
There are demonstrators in these labs to help you

These are ***quiet*** labs

The web



Most of the resources you will need can be found on the CompSci 101 website:

`http://www.cs.auckland.ac.nz/compsci101s1c`

21

What is an algorithm?

A finite set of steps that specify a sequence of operations to be carried out in order to solve a specific problem.

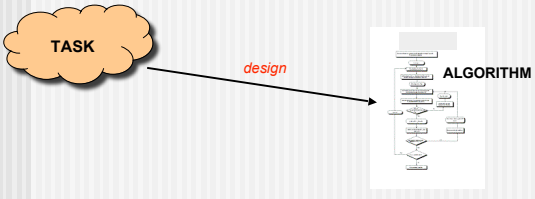
Can't use our natural language

"Pick a secret word and allow the player to select one letter at a time until the entire word is guessed. The player should be informed each time a correct letter is guessed."

We need to give detailed instructions using language that the computer understands.

23

Programming Step 1 - Design the algorithm



24

Algorithm

An algorithm is a well-defined, unambiguous sequence of steps

sequential operations

Walk to the bus stop at the shops down the road
Get on bus number "101"
Pay the bus driver $3.20
Get off at the Symonds St bus stop
Walk 200m to the Computer Science building

Algorithms

HOW TO WALK TO THE BUS STOP

25

The steps can include conditional operations

Open the front door
IF it is raining **THEN**
  take an umbrella
Walk down the driveway and turn left
Walk 50m down the street

conditional operation

Algorithms

HOW TO PAY THE BUS FARE

26

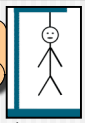
The steps can include iterative operations

Open wallet
WHILE you still haven't paid enough
  give the driver another coin

iterative operation

Hangman algorithm

TASK



hangman

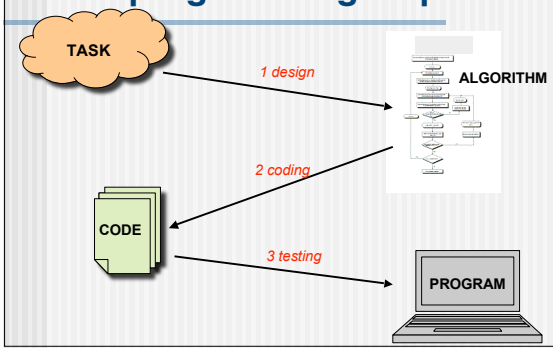
sequential operations
(1) generate a secret word
(2) display a "_" for each character in the word
(3) set "misses" to 0
(4) wait for the user to guess a letter

conditional operations
(5) IF the letter is in the word
  replace the corresponding "_" with the letter
otherwise
  add 1 to "misses"
(6) IF the whole word is guessed THEN
  display "Well Done" and END

iterative operations
(7) GO BACK to (4)

Basic programming steps

28



Programming Step 2 - Write the code

29

We can use the Java programming language to implement our algorithms