

# Achievement project for AS91074

Demonstrate understanding of basic concepts from computer science

This project is intended for students who have limited time for AS91074, and/or who just want to aim for an achieved so that they get the credits. The project only covers the achieved level, although is intended to still be interesting and meaningful, and puts a strong emphasis on carrying out practical activities rather than doing large amounts of reading, writing, and analysing.

## A word of warning

**If you don't have much time for this standard, you may find it tempting to copy or paraphrase information from wikipedia, notes from your teacher, or other sources. While you may think this will help you achieve the standard, the markers do not want to see what you can copy paste or paraphrase, and it does nothing to show your own *understanding*. The markers would like to see you carry out actual activities and then report on what *you* did, and what *you* learnt from them.**

**Also, report length is unimportant. Less than 10 pages of high quality work is far better than 10 pages of mixed quality work which was deliberately bulked out.**

## Requirements for the standard

Taken from <http://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2011/as91074.pdf>

- describe the key characteristics and roles of algorithms, programs and informal instructions.
- describe an algorithm for a task, show understanding of the kinds of steps that can be in an algorithm, and determine the cost of an algorithm for a problem of a particular size.
- describe the role and characteristics of programming languages, including the different roles and characteristics of high level languages and low level (or machine) languages, and the function of a compiler.
- describe the role of a user interface and factors that contribute to its usability.

## Referencing of Sources and plagiarism

It is a NZQA requirement that you reference **all** sources of information you use that were not yours. In particular, this includes examples supplied by your teacher, such as a low level program, binary search algorithm, and binary search program.

While you are very unlikely to need to include them in your report, notes/ images given to you by your teacher or text/ images from websites or books must also be clearly referenced.

It is not sufficient to include a list of sources at the end of your report. You must show clearly within your report what information was taken from where. This can be done with a sentence such as “My teacher gave me this example”.

If you are unsure about whether or not what you are doing is okay, there is a brilliant powerpoint document by Azman and Fox from the University of Hawai'i.

[http://turnitin.com/assets/en\\_us/images/lesson-plans/20140421-plagiarism-seuss.ppt](http://turnitin.com/assets/en_us/images/lesson-plans/20140421-plagiarism-seuss.ppt)

## Project Overview

This project comes along with a writing frame for the report that you will need to submit. The writing frame also provides some sentence starters.

The writing frame can be found [\[here\]](#).

The project is split into 4 parts: Algorithms 1, Algorithms 2, Programming Languages, and Human Computer Interaction. Algorithms 1 and Programming Languages are straightforward knowledge tasks, where you will show the marker that you understand the concepts you learnt in class by relating them to examples. Algorithms 2 and Human Computer Interaction will require you to carry out (hopefully interesting) mini projects and report on them.

# Algorithms 1

As long as you understood the concepts that were introduced in class, this part should not take you long.

If you wanted your friend to look up the phone number for the local pizza shop so that you could order a pizza, what might you say to them so that they know you want them to do this? Your answer to this question is an example of an informal instruction.

In your report:

- In the boxes on the writing frame which are provided for your examples:
  - Include your *informal instruction* example that you just came up with (to get your friend to order a pizza).
  - Include the binary search *algorithm* example your teacher gave you. **Make sure you have referenced it by stating it was teacher provided.**
  - Include the binary search *program* example your teacher gave you. **Make sure you have referenced it by stating it was teacher provided.**
- Beneath the examples, you will need to describe
- Answer a few questions in your report about the examples, to show that you understand what an algorithm, program, and informal instruction is. Some useful sentence starters are provided in the recommended layout. These will help ensure you link your explanations back to the examples. Note that it is not sufficient to copy paste material from the web or your class notes and exercises into your report. It must be your own words, and linked to the examples.
  - Why would your informal instruction be suitable for a human, yet unsuitable for a computer?
  - What is at least one difference between your algorithm and program?

## Algorithms 2

For this activity, you will need a phonebook and a camera (a smartphone camera is fine).

In class, you learnt about **Binary Search**, and about how it can be used on a phonebook. For this part of your project, you will show that you understand how to carry out a binary search by finding the phone number for your favourite shop in a phone book and showing the steps you took to do so. You will also keep track of how many steps you took, which is the cost of the binary search algorithm for the problem of finding your favourite shop.

Before you go any further, start by explaining in your own words what it is that you going to be doing. Be sure your explanation includes the name of the shop you are going to be searching for. This explanation should be around 2 or 3 sentences, and will be used as an introduction to this section.

Once you have written what it is you are going to be doing, you can start carrying out your binary search. Take a photo of each page you have to look at while carrying out the binary search. Just take a photo of the single pages which were part of the search (i.e. don't also photograph the adjacent pages). You will be putting these photos in your report, so ensure there is sufficient light and that the photos are properly focussed.

In your report you should include:

- The intro you wrote that explains what it is you are doing (this goes just after the heading).
- In the large table on the writing frame you should:
  - Insert the photos for each step of your search.
  - Underneath each photo, explain what decision you made and why (e.g. how did you decide whether the target must be before or after this page?). State what range you have narrowed the search down to.
- Explain what the cost of the algorithm was for *your* example, and how you got to this answer. Also, determine what the cost for your example would have been if you had used linear search (this is very easy). Some useful sentence starters are provided in the writing frame.
- Explain *what you have learnt* about binary search from searching the phonebook for the shop, and why it seems better than linear search.

Note: You should use the page numbers (and use a calculator if necessary) each time you need to find the “middle” page during your binary search. However, don't worry about being “off by 1”. For example. if you are finding the page between 58 and 71, then it could be either 64 or 65, because there is no 64.5. It does not matter which you choose.

The number of pages you need to look at will depend on the size of the phonebook, and whether or not you get really lucky and find it very early in your search. As a general rule though, if you have to look at more than 13 or 14 pages, you have probably done something wrong and should either try again or ask your teacher to check that you are doing the search correctly. Looking at somewhere between 6 and 12 pages would be the most likely.

# Programming Languages

As long as you understood the concepts that were introduced in class, this part should not take you long.

Note that it is not sufficient to copy paste material from the web into your report. It must be your own words, and linked to the examples.

In your report:

- Find a program that you wrote. The program should contain at least one loop and at least one if statement. Trim it down to less than 15 lines (around 10 is ideal) and include it in your report. This is your high level language example.
- Your teacher will give you a second program which is in a low level language. This is your low level language example. Reference it as teacher supplied.
- For both the programs, briefly describe the output. This is just to prove that you know what the programs you have included in your report do. Your teacher will have told you what the low level program does.
- Answer a few questions in your report about the examples, to show that you understand what a low level language, high level language, and compiler is. Some useful sentence starters are provided in the recommended layout. These will help ensure you link your explanations back to the examples.
  - What are two features of the programming language used in your high level language example which make it easy for humans to understand? (you might like to use your low level language example as a comparison).
  - Why would you not want to program in the low level language?
  - Why do computers require low level languages?
  - How does a compiler solve the problem of differing human and computer needs?

# Human Computer Interaction

Human Computer Interaction is all about making the interface (the part of the app or program that you can see) as easy to use as possible. A great interface will be easy to use, even for somebody who has never seen it before, and perhaps who doesn't even use computers very often. For the final part of your project, you are going to see what happens when one of your grandparents (or another person who doesn't tend to carry electronic devices around) attempts to carry out a simple task on one of your electronic devices. The person you choose will be referred to as your "helper".

Choose **one** of the following tasks: (You may need to fill in the blanks for your chosen option).

1. Ask your helper to use your cellphone to send the text message [\_\_\_\_\_]  
to your friend [\_\_\_\_\_]. The cell phone should start at its home screen, unlocked.
2. Ask your helper to use your mp3 player to play the song [\_\_\_\_\_]. The mp3 player should start at its home screen.
3. Ask your helper to use your cellphone to call your emergency contact called "Mum". The phone should start (and stay) in its "locked" state - the idea of this feature is that if somebody finds you unconscious on the street, they are able to contact somebody who knows you, even if they cannot unlock your phone. Note that this one requires some additional setup. Set an emergency contact called "Mum" (but it can just be set to a friend's number if you want - for the purpose of this report). Also make sure your phone has a password so that it cannot be unlocked by the helper.

The person I have chosen to be my helper is [\_\_\_\_\_]

Hand the device to your helper and tell them what their task is. Ask them to talk out loud and explain why they are doing what they are doing (important). Do not tell them anything else about the task, although you might need to explain why you are asking them to do this.

While they are trying to carry out the task, you should be keeping a list of every step they try, including what they are saying. If you are able to (e.g. have a digital camera or cellphone that is not currently being used), you might like to video your helper carrying out the task so that you can refer back to it later, as this will help you to write your report. The list of steps should be of a high enough quality to include in your report.

In your report:

- Fill out the brief introduction to what your interface was, who your helper was, and what the chosen task was.
- Explain what the purpose of your chosen interface is.
- Include one or two photos of your chosen device.
- Put the steps that your helper took into your report.
- Fill out at least 2 paragraphs in the analysis section. These are further explanation about some parts of the task that your helper found really difficult (or easy), and how the design of the interface contributed.

# Submission Checklist

## Algorithms 1

- I have included my informal instruction (how I would ask a friend to search for a phone number in a phonebook).
- I have included the example of an algorithm for binary search that my teacher gave me.
- I have included the example of a program for binary search that my teacher gave me.
- I have *referenced* the binary search algorithm as teacher supplied.
- I have *referenced* the binary search program as teacher supplied.
- I have explained why my friend can easily make sense of the informal instruction.
- I have explained why the informal instruction I'd give my friend would be unsuitable to give to a computer.
- I have explained what the different roles of the program and algorithm are.
- I have pointed out an example of an if statement being used in the binary search algorithm.
- I have explained why that particular if statement is needed by the binary search algorithm.
- I have pointed out an example of a loop being used in the binary search algorithm.
- I have explained why that particular loop is needed by the binary search algorithm.

## Algorithms 2

- I have carried out my binary search and taken photos of each step.
- I have explained how I carried out my binary search.
- I have included the steps of my binary search on a phone book.
- I have explained (briefly) what I did in each step.
- I have specified the cost of the binary search algorithm for the search I carried out, and described how I got that answer.
- I have specified what the cost would have been if I had used linear search.
- I have explained what I have learnt about binary search and linear search from doing the exercise.

## Programming Languages

- I have included an example of a high level program that I wrote.
- I have included the example of a low level program my teacher gave me.
- I have *referenced* the low level program as teacher supplied.
- I have explained at least two specific features of the high level program which make it easier for a human to understand.
- I have explained at least two specific features of the low level program which make it challenging for a computer to understand.
- I have explained why computers need low level languages.
- I have explained how compilers solve the problem of humans liking high level languages and computers liking low level languages.
- I have ensured that all my descriptions are my own, based on what I learnt in class, and not copied or paraphrased from Wikipedia, other websites, or teacher notes given to me while learning.

## Human Computer Interaction

- I have stated what my interface was.
- I have stated who my helper was.

- I have stated what the chosen task was.
- I have explained what the purpose of my chosen interface is.
- I have included a photo of my interface.
- I have included the list of steps my helper took when attempting the task.
- I have explained at least two aspects of the interface which my helper found difficult (or easy).

### **General**

- All my images are legible, even when printed or in their final submission form.
- I have read over my report to make sure there are no incomplete sentences or obvious errors.
- I have removed *all* the red guide notes from my report.

### **Last check before submitting**

- I have added my NSN number as a header to my report.
- I have double checked that I have removed *all* the red guide notes from my report.
- I have double checked that there is no random whitespaces or obvious formatting errors in my report.
- I have double checked that images/ examples which are not my own are suitably referenced.
- I have double checked that my images are all where I want them.