Digital Technologies 3.7: AS 91906v1

Assessment Task

Use complex programming techniques to develop a computer program

Level Three, Credits 6, Assessment Internal

This achievement standard involves using complex programming techniques to develop a computer program.

Achievement	Achievement with Merit	Achievement with Excellence
Use complex programming techniques to develop a computer program.	Use complex programming techniques to develop an informed computer program.	Use complex programming techniques to develop a refined computer program.

A copy of the standard can be found here:
https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91906.pdf

Conditions/Ngā Tikanga

In this assessment you **must** not consult with other students. You **may** consult your own notes.

Assessment misconduct and breaches of the rules

Assessment misconduct is failing to obey the conditions under which an assessment activity is run.
It takes many forms:

- copying material from a source like a book, the internet or other electronic material without acknowledging that the material, words or ideas are someone else's. A few word changes don't make this form of plagiarism any more acceptable.
- undue help from home, or from a tutor, or from a reader-writer
- using or copying all or part of another student's work
- giving work to another student which allows that student to use it or copy it
- using written or recorded material in an internal assessment or NCEA examination
- using language dictionaries, electronic spell checkers, unauthorised calculators, iPods, smartwatches or cell/smart phones in all assessments
- disrupting an examination, e.g. moving around the room, scanning neighbour's papers.
- failing to follow instructions during the internal or external assessment

Involving other students

Where a second student has participated in the misconduct, that person will be penalised also.
It is very important that you don't lend your assessment work. It is okay to discuss the topic you are doing with someone else but the final piece of work must be your own.

Student/Ākonga instructions

Read these instructions carefully.

This assessment activity requires you to create a computer program, using complex techniques.

A complex computer program:

- uses variables storing at least two types of data (e.g. numeric, text, Boolean, object)
- uses sequence, selection and iteration control structures
- takes input from a user, file, sensors, or other external source
- produces output
- uses two or more complex programming techniques.

Examples of complex programming techniques include:

- programming or writing code for a graphical user interface (GUI)
- reading from, or writing to, files or other persistent storage
- object-oriented programming using class(es) and objects defined by the student
- using types defined by the student
- using third party or non-core API, library or framework
- using complex data structures (e.g. stacks, queues, trees).

You will be assessed on how effectively you develop, test and refine your program, so that it is a well-structured, logical response to the task. While developing, testing and refining your program:

- write code that meets all the task specifications
- set out the program code clearly, following conventions of your chosen programming language
- document the program with appropriate variable/module names and organised comments that describe code function and behaviour
- comprehensively test and debug your program in an organised way, to ensure that it works on a sample of both expected cases, relevant boundary cases, and invalid cases
- ensure that the program is a well-structured, logical response to the task
- make the program flexible and robust.

Time Allowances

You will be given **2** weeks of class time.

There will be absolutely **NO** extensions.

Problem Statement

In 2020, the rapper Logic spent US$226,000 on a Pokémon Card (Espinoza, 2020). You have decided that it would be handy to have a program that stored your collection of Pokémon cards and their monetary value.

Your code needs to be able to:

- add a Pokémon card's name, monetary value and picture;
- get a Pokémon card, show its name, monetary value, and picture.

Details

What your code needs to do

Your code should:

- allow the user to add a new Pokémon card;
- allow the user to search a Pokémon card to get their details – the name, monetary value, and the image should all be shown;
- allow the user to click a button to hide the Pokémon card's details;
- allow the user to click on the image to hide the current Pokémon card's details;
- allow the user to click a button to see the names and monetary values of all the Pokémon cards in the collection.

Reference

Espinoza, J. (2020). *Logic Reportedly Dropped $226,000 on Pokémon Card.* Retrieved from https://www.complex.com/music/2020/10/logic-spends-226-thousand-dollars-on-rare-pokemon-card

Notes

- You must document the development process.
- Your program must include comments. You should complete the `README.md` file for the project as well.
- You must provide evidence of how you tested your program.
- You must explain what you did to debug your program, if your testing showed evidence of bugs.

Submission

When you have finished your program and its testing, check that you have met the requirements of the marking schedule.

IMPORTANT

- There will be successive submission drops made available. Make sure you utilise these drops for version control and in case of disaster recovery;
- You must record your progress for each day using the litedocumentation provided;
- You must sign and return the authenticity declaration.

Make sure you turn in the work you want to be marked in the final submission drop.

The **final** version, should contain **only** what you want to be marked:

- Your source code
- Your `README.md`
- Your testing in `.docx` format
- your litedocumentation in `.docx` format
- your authenticity declaration in `.docx` format

Pokemon Card Collection Plan

Card.java

Constructor:

- String Name
- Int Value
- String ImgFile

} set all values for each card instance

Getter Methods:

Collection.java

CollectionGUI.java