### **Title Page**

Please make sure your title page contains the following information.

**Title**

Student Generated Videos

The title should be concise and informative.

**Author information**

Adam Ka Lok Wong  
College of Professional and Continuing Education, The Hong Kong Polytechnic University, Hong Kong 100077, China

[adam.wong@cpce-polyu.edu.hk](mailto:adam.wong@cpce-polyu.edu.hk)

* *The name(s) of the author(s)*
* *The affiliation(s) of the author(s), i.e. institution, (department), city, (state), country*
* *A clear indication and an active e-mail address of the corresponding author*
* *If available, the 16-digit ORCID of the author(s)*

*If address information is provided with the affiliation(s) it will also be published.*

*For authors that are (temporarily) unaffiliated we will only capture their city and country of residence, not their e-mail address unless specifically requested.*

**Abstract**

### Background

### Methods

237 students submitted their individual computer programming assignment for assessment. There are three parts to the assignment. The first part is the write a C++ program using a template file. The second part is to create a video of their program for Question One of Part 1. The third part is to evaluate other students’ videos based on the given criteria and rubrics. Analyses were conducted for Part 2 and 3 of the assignment.

### Results

### Limitations

### Conclusions

By providing a separate set of instructions and watching demonstration videos before students doing peer evaluation are conducive, as students have gained a fuller understanding of the marking criteria and rubrics, hence can provide more accurate evaluation.

*Please provide an abstract of 150 to 250 words. The abstract should not contain any undefined abbreviations or unspecified references.*

For life science journals only (when applicable)

* *Trial registration number and date of registration for prospectively registered trials*
* *Trial registration number and date of registration, followed by “retrospectively registered”, for retrospectively registered trials*

#### **Keywords**

Please provide 4 to 6 keywords which can be used for indexing purposes.

#### **Statements and Declarations**

The following statements should be included under the heading "Statements and Declarations" for inclusion in the published paper. Please note that submissions that do not include relevant declarations will be returned as incomplete.

* **Competing Interests:** Authors are required to disclose financial or non-financial interests that are directly or indirectly related to the work submitted for publication. Please refer to “Competing Interests and Funding” below for more information on how to complete this section.

Please see the relevant sections in the submission guidelines for further information as well as various examples of wording. Please revise/customize the sample statements according to your own needs.

# Background

This write-up is about the individual assignment of SEHH2042 Computer Programming of Hong Kong Community College (HKCC) of The Hong Kong Polytechnic University (PolyU). This is a 2nd year level course of the associate degree programme. There are 255 students in this course, and 237 students have submitted the assignment for marking. The Expected Learning Outcomes are to familiarize themselves with at least one high-level language programming environment, develop a structured and documented computer program, and apply the computer programming techniques to solve practical problems.

# Methodology

**Week 5: Part 1 of the assignment**

Part 1 of the assignment is worth 80% of the total assignment mark. There are four questions to Part 1. Students are to write a C++ program. They are given a C++ program template file and are required to insert C++ codes into the template file according to the given instructions. The final program should be able to satisfy all requirements listed in the notes.

The instructions include inserting codes into the functions as specified in the template file. To answer question 1, students need to write their code in the scope of Q1 ().

When the program is executed, enter the question number to run the code of a particular question. They may write user-defined functions for solving question 1, and call it the in the given function Q1(). They can include more header files in the template file if necessary. Apart from inserting codes as mentioned above, they cannot modify a given code in the template file. They may assume that users always provide valid input. No error input checking is required unless required by the question.

**Week 5: Part 2 of the assignment**

Part 2 of the assignment is worth 10% of the total assignment marks. Students are to create a screencast (video) demonstrating how they run their code from Part 1 of Question 1 to test all important scenarios. The video should also include their voice recording in English as they explain what they are doing at each step, such as what scenario they are testing and what results are printed on the screen as they enter the input integers.

The video requirements include the video does not show the 'showInfo' function in their video. The video must show the whole screen, not just part of the screen. The source code of Part 1 of Question 1 needs to be shown clearly. The video must be saved in a .mp4 or .mov file. Last but not least, the duration of the video cannot be longer than 3 minutes. Students are allowed to use any software to create their screencast. Screencastify is available in Moodle for students' reference.

**Week 7: Part 3 of the assignment**

Part 3 of the assignment is worth 10% of the total assignment mark. To conduct peer-review by evaluating screencasts created by other students. Students are given screencasts of four other students and to watch and evaluate them based on the following criteria and rubrics. The marks that one awards will not actually be the marks that are awarded to the students who created these screencasts. However, the marks that a student awards will be used to the determine the mark that one scores in Part 3, depending on how accurately one marks the screencasts (compared with the marks that the teachers award).

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **2 marks** | **1 mark** | **0 marks** |
| **Demonstration of testing** | All important scenarios are adequately tested | Some (at least one) important scenarios are not adequately tested | Too many important scenarios are not adequately tested |
| **Communication clarity** | Everything is clearly communicated, using both actions on screen and voice | Some ideas are not clearly verbally explained or shown clearly on the screen | Too many ideas are not clearly verbally explained or shown clearly on the screen |

# Findings

The first observation of this study is that students are instructed to save their video assignments under their student identification numbers and student names. In this connection, fellow students may tend to give higher marks if they mark their friends' videos. Please see Figure 1 and 2. (Insert Figure 1 and 2)

The second observation is that not all students provide the same amount of information in the videos. Some students provide additional information like what the assignment is about and the requirements, some run the computer program right from the beginning without explanation much what they are doing, and some with lengthy explanations but do not run any computer programs, so they do not score any points for demonstration of testing.

The third observation is that there is inconsistency between the marks of teacher’s and peer evaluation. The main reason is that students do not have a clear understanding of the rubrics, so they tend to give higher marks when watching a high-quality video and give lower marks if watching average-quality videos. However, to score 2/2 in demonstration of testing, one only needs to run all of the required scenarios and explain what scenario they are testing, and what results are printed on the screen as they enter the input integers. To score 2/2 in communication clarity, one only needs to follow 100% of the video requirements. However, the grades for peer evaluation are quite diverse. In addition to whether the program is run and explained in the videos, students tend to give marks based on how clear and adequately the demonstrations of the program is explained, including the quality of English and how much additional information students provide in the videos. In fact, English is not part of the marking criteria as students only need to explain what scenario they are testing and explanations need not be lengthy. Also, the use of subtitles and diagrams will not lead to higher marks by the teacher. Some students have spent time making diagrams and subtitles as shown below. Please see Figure 3 and 4. (Insert Figure 3 and 4)

Another reason for the inconsistency is that students do not that four scenarios are required in order to get 2/2 in demonstration of testing, students must run four scenarios instead of three. Students usually give 2/2 to the videos if they see three scenarios with explanations and results are printed on the screen and do not know the requirement of the four scenario. This is done because the lecturer is doing it purposefully to test students' understanding of the question.

Another observation is that not all students use the same program to do the assignment. Students should be asked to use the same program, e.g., Visio Studio, to standardize the marking and make it fair to everyone. Some students may favour the interface of one program over another as it may appear to be fancier to the peer evaluator. Please see Figure 5 and 6. (Insert Figure 5 and 6)

Lastly, it is noted that some students speak perfect English in the videos, but the evaluator cannot determine if they are really done by the students themselves.

# Suggestions

There are two ways to narrow the gap between the marks between teacher's assessment and peer evaluation. The first way is to provide a second set of instructions to be given to students before doing peer evaluation, on Week 6, after submitting Part 1 and 2 of the assignment.

This way, students know what to note when conducting peer evaluations and can look for awarding and deducting points.

The second way to narrow down the gap between the marks given by the teacher and peer evaluation is to ask students to watch some demo videos before peer evaluation. This way, they know exactly what a good video and an average video are like. As this study will be continued in Semester 2, it is suggested that the teacher selects one video from Semester 1 that gets 2/2 in both demonstrations of testing and communication clarity and one video that gets ½ in both demonstrations of testing and communication clarity.

One way to prevent favoritism in peer evaluation is be asking students to blur the filename or anywhere that shows students' identities in the video.

Afterwards, the teacher then assigns individual videos to students for peer evaluation. Even though this procedure is tedious and time-consuming, it can prevent favoritism in peer evaluation.

We can make the instructions and rubrics more precise students know what to include in the videos. More details can be added for the rubrics to understand what they need to do to get full marks or get the mark they desire. Students would not add the diagrams and subtitles if the rubrics reflect that adding these would not lead to a higher mark in communication clarity. In addition, all students will run the programs in the video instead of just explaining what they have done without running the program.

To prevent students from becoming discouraged, we can change the marking scheme slightly by awarding 3/3 if all four scenarios are demonstrated and deducting one point for each scenario missing. Currently, one point is deducted if only three scenarios are demonstrated and one point is deducted if any of the video requirements are missing. This way, students would score 50% on the assignment after putting in so much effort and can be discouraging. Increasing the total marks from 2 to 3 would ensure students getting a reasonable score even missing one scenario and some video requirements.