Name_____ Date_____ Block_____

IB Computer Science SL/HL

Internal Assessment

Project Overview

For your IB Computer Science Internal Assessment, you will be tasked with the development of a large-scale, software development project in which you will independently design and develop a fully working application, designed to meet the needs of a clearly identified client and/or end-user.

Choice of Topic

In identifying a problem, you are free to select any topic that appeals to your interests. However, you should undertake a task that is challenging enough and requires the application of techniques that will allow you to showcase your programming, problem-solving, and organizational skills. It is essential for your solution (i.e., your finished program) allows you to explicitly demonstrate and document your algorithmic thinking skills.

Choice of Client/Adviser

Over the course of the project, you will need to work closely with your selected client throughout the development of the solution. Therefore it is recommended that wherever possible, you select an adviser who is known to you or your family. This could include members of the school community, local clubs, or businesses.

Requirements

The finished deliverables for the overall project will include:

Cover Page A web page containing relative links to the product and its associated documentation

The Product A fully working software application

Documentation Extended writing, diagrams, pseudocode, and other evidence documenting your design, development, and testing process
A video demonstration showing off the full functionality of your program

Students should aim to develop a ***product*** that uses appropriate (i.e., *complex, non-trivial*) techniques, is fully functional, and allows access to its complete internal structure for assessment purposes. The video should show the product functioning and should address the ***success criteria*** stated during the planning phase of the project.

Project Templates

At the start of the project, students will be provided with a set of files to be used as the basis of their submission.

- 🗁 assessment
  - ↳ 🗋 AssessmentByStudent.docx — *Use this to **self-assess** your work as you progress through the project.*
  - ↳ 🗋 AssessmentByTeacher.docx — *To be completed by the teacher at the conclusion of the project.*
  - ↳ 🗋 Form_4Compsci_e.pdf — ***Fill out, print, and sign** this form when submitting your final project.*
- 🗋 cover_page.html — ***Edit this file** in a text editor to include your name, id, project info, etc.*
- 🗋 demo.mov — ***Record a video** demonstrating the full functionality of your product.*
- 🗁 documentation
  - ↳ 🗋 A_Planning.docx — *Write an analysis of the problem to be solved. (1 page)*
  - ↳ 🗋 B_Design.docx — *Document the design and structure of your project. (2-3 pages)*
  - ↳ 🗋 B_RecordOfTasks.docx — *Construct a detailed schedule of tasks before starting each stage.*
  - ↳ 🗋 C_Development.docx — *Document your development process. (1-2 pages)*
  - ↳ 🗋 D_Functionality.docx — *Document key features and extensibility of your project. (x pages)*
  - ↳ 🗋 E_Evaluation.docx — *Write an assessment of the success/failure of your project. (1 page)*
  - ↳ 🗋 X_Appendix.docx — *Include any supplemental materials referenced in Parts A-E.*
- 🗁 images
  - ↳ 🗋 ib_faded.png — *Background image for the HTML cover page.*
- 🗁 product — ***Include** all source code files, libraries, and executable JAR file here.*
- 🗋 README.txt — ***Edit this file** in a text editor to include your name, id, project info, etc.*

Part A: Planning

The Client *20-30 words*

Provide a brief statement identifying the client in terms of the type of end-user for whom you are developing this program.

The Problem *15-20 words*

Provide a one-line statement describing the type of problem to be solved.

Current Solution *100 words*

Provide a detailed description of the solution(s) currently available to the client, including the processes they must go through as well as the advantages and disadvantages of the current system.

Sample Data *40-100 words, not including data*

Provide specific examples (collected during your client interviews) of the various pieces of data and explain how they will be used in the context of your program.

Analysis of the Problem *100-150 words*

Thoroughly describe, analyze, and evaluate the challenges that you will need to address in developing a successful solution to your client's problem.

The Proposed Solution *75-100 words*

Provide a thorough and detailed description of the proposed solution and justify why/how your product will meet the needs of the client.

Criteria for Success *1 list item per feature*

Create a bulleted list of the features that are essential to any program of this type.

Create a bulleted list of the basic features that are requested specifically by the client.

Create a bulleted list of the optional features that might enhance the program beyond the client's requirements.

Part B: Solution Overview

Design Overview *50-100 words*

Briefly describe how you approached designing a product that will meet the needs of your client.

Algorithms *1 table per feature*

Identify all of the key functionality that your program will need to implement and use the provided tables to describe the necessary specifications (i.e., preconditions and postconditions) and basic logic (i.e., pseudocode) for each. Add additional tables as necessary.

User-Defined Data Structures *1 table per class*

Identify all of the key data structures that your program will need to implement and use the provided tables to describe the stored data (i.e., data fields) and behaviors (i.e., methods) for each. Add additional tables as necessary.

File Structures *1 table per file type*

Identify all of the data files that your program will use to store information between sessions and use the provided tables to describe how the data is organized when it is written into or read from the files. Add additional tables as necessary.

Test Plan *1 table entry per feature*

Identify all key features of your program and design test cases to thoroughly test the functionality and robustness of your program. Be sure to include all general cases as well as all boundary cases and special cases, including good user input and bad user input. Your program should be able to handle all good input correctly and all bad input gracefully (i.e., without crashing or misbehaving).

User Interface Design *1 image per screen*

Construct a short narrative of a typical user experience when using your program, including scanned images/sketches depicting mock-ups of screens, input controls, and output displays for each part of the intended user interface.

Part B: Record of Tasks

NOTE: This task list should be developed at the ***start*** of the project as a way ***plan*** out all phases of the development process prior to beginning any work on the project. You are advised to be as ***thorough*** as possible and to set ***reasonable*** "Target Dates" that will allow you to stay on schedule and still meet the established due date.

Analysis Phase	*1 item per task*

Identify and list all of the tasks necessary for selecting and meeting with a client, gathering information about the problem, and establishing your overall goals for the project. Add additional rows to the table as necessary.

Design Phase	*1 item per task*

Identify and list all of the tasks necessary for designing the overall look and behavior of your program such that it meets the client's needs. Add additional rows to the table as necessary.

Implementation Phase	*1 item per task*

Identify and list all of the tasks necessary for implementing all of the necessary classes and data structures that will make up your program. Add additional rows to the table as necessary.

Testing Phase	*1 item per task*

Identify and list all of the tasks necessary for thoroughly testing the functionality and robustness of your program. Add additional rows to the table as necessary.

Deployment Phase	*1 item per task*

Identify and list all of the tasks necessary for documenting, packaging, and distributing your program and preparing it for delivery to IB for moderation. Add additional rows to the table as necessary.

Maintenance Phase	*1 item per task*

Identify and list all of the tasks necessary for future use, maintenance, and enhancements to your program after delivering a working solution to your client. Add additional rows to the table as necessary.

Part C: Development

Development Process *50-100 words*

Briefly describe your development and implementation processes and methodology.

Product Structure *300-600 words*

Model

Describe the "model" components/classes and the roles they play in your program.

View

Describe the "view" components/classes and the roles they play in your program.

Controller

Describe the "controller" components/classes and the roles they play in your program.

Algorithmic Thinking *100-200 words + annotated screenshots*

Provide specific examples from throughout your program that highlight your ability to apply algorithmic thinking to the implementation process.

Development Tools *50-100 words*

State the programming tools (i.e., IDE, language, SDK, etc.) used to develop your solution and briefly explain why they were an appropriate choice for this task.

Libraries *1 description per library*

Use the provided table to list each of the SDK libraries used throughout your project and describe the role that each served in the development of your program.

External Resources *1 citation per resource*

Include all acknowledgments and citations for any 3rd-party templates, program snippets, or other materials that have been used or modified in your program.

Part D: Functionality and Extensibility

Key Functionality *1 list item per feature*

Create a bulleted list of all key features of your program that are demonstrated in the video. Specifically, this list should include all items included in the "Criteria for Success" identified in Part A as well as any additional features that you have incorporated into your program.

Video Transcript *does not need to be verbatim*

Include a transcript of the information provided in the video. This can either be transcribed from the video after it is shot or it can be written before hand and can serve as a script to read as you walk the view through your program.

Video Demonstration *3-6 minutes*

Record a short video that thoroughly details the full functionality of your finished program. Guide the viewer through the proper use of the program, highlighting each of its key features, including the ability to appropriately handle both good and bad user input.

Recording Process

The use of a screen-recording application to record the actual mouse movements and on-screen elements as well as a vocal narrative is recommended. Alternatively, using a tripod-mounted camera to externally film the screen is permissible, but holding the camera by hand is discouraged.

A few of the recommended screen-recording applications (available for Mac or PC) include:

- *Debut Video Capture* (http://www.nchsoftware.com/capture/)
- *QuickTime* (https://www.apple.com/quicktime/download/)

Also, as you run and record your program, be aware of (and close) any other visible distractions that might show up in the video (i.e., notifications, chat boxes, open folders, files, other applications, etc.). Your aim should be to create an informative and professional-looking demonstration of your program.

Video Formats

You may save your video in any common digital video format (e.g., .mp4, .mov, .mpg, .avi, etc.). For example, if you can upload it to YouTube, it's a suitable format.

Size and Resolution

You should ensure that the resolution of your video is high enough such that any on-screen text is clearly readable. If you are using screen-recording software, recording at your display's native resolution is usually recommended, but anything over 720p is most likely overkill (and will result in unnecessarily large files).

Extensibility of Product *50-100 words*

Identify ways in which the design of the program allows for easy maintenance and/or modifications by the end-user or future developers (e.g., modular design, data structures, descriptive naming conventions, comments, and documentation).

Part E: Evaluation

Product Evaluation *150-200 words*

Evaluate the success of the final product in terms of the "Criteria for Success" identified in Part A and note how each feature supports the client's needs.

Client Feedback *50-100 words*

Document all feedback, comments, suggestions, and additional feature requests from the client and/or end-users.

Future Development *175-250 words*

Describe any possible enhancements and/or additional features that you or your client might recommend to improve your program in the future.

Appendix (Optional)

Supplemental Materials

Include any additional materials that you might have referenced elsewhere in Parts A – E.