

**INSTRUCTIONS:**

**Issue Date:** 18/7/2018

**Implemented System:** 31/8/2018

**Documentation:** 7/9/2018

**Conditions:**

Time allowed: 7 weeks

You have seven weeks to produce a working program using **Python**.

This task is to be completed during class and in your own time. The teacher will act in a consultative role.

This question sheet, together with the criteria sheet, must be attached to your assignment.

Your implemented system must be submitted to Moodle on the due date.

This criteria sheet (signed) must be handed into **Student Services** by 8.15am on the due date, and the documentation and product uploaded to **LMS**.

|  |  |
| --- | --- |
| **Criteria** | **Standard** |
| Criterion A: KA |  |
| Criterion B: AS |  |
| Criterion C: EC |  |
| **Overall** |  |

**IPT**

**Software Development**

#### **YEAR 11/12**

**Semester 2/4**

**Major Project – Item D**

**Assessment Techniques:**

* Product Major Project
* Individual
* Yr 11: 800 – 1000 words + Phyton program
* Yr 12: 1000 – 1500 words + Python Program

**Criteria Assessed:**

* Knowledge and Application
* Analysis and Synthesis
* Evaluation and Communication

**Common Curriculum Elements:**

* Analysing
* Synthesising
* Explaining to others
* Applying a progression of steps to achieve the required answer
* Applying strategies to trial and test ideas and procedures

**Student Name:**

|  |
| --- |
| **STATEMENT OF AUTHORSHIP:**  I accessed the following resources during the stages of text production (please tick):   * Print resources: fiction, non-fiction, periodicals/magazines, newspapers, other * Digital resources: websites, YouTube, Podcasts, film/music excerpts, other * Human resources: teacher, librarian, tutor, parent/guardian, peer/s, other   Other than the above sources, which I have correctly acknowledged, the work I have submitted is entirely my own.  Student’s signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |
| --- |
| **TASK** |
| You task is to design, develop and evaluate one working game using **Python 3.6**. Three common games have been provided. The projects range in difficulty from more basic to more challenging. Choose a project that you feel confident that you can design and implement. |

|  |
| --- |
| **DOCUMENTATION** |
| You are required to submit two documents. |
| **You are required to submit a word-processed specification document including:**   * Title Page * Table of Contents * Problem Definition * Program Objectives * Testing Plan * UI Designs * List Data types and functions * Flowchart or pseudocode * Prinout of code * Testing Documentation – Test Data, Test Cases, Problems and 3rd Party Reports * Evaluation * Project Log and signed Declaration |
| **You are required to submit a user guide document including:**   * Introduction - what your game does * System requirements for the game * How to install the game * How to play the game |

|  |
| --- |
| **GAME OPTIONS** |
| **Option 1 – Primary Maths Game**  Create a computer game to help students learn mathematics.  The game quizzes the user on mathematic questions. The user should be able to choose which of the four operators will be quized (addition, subtraction, multiplication and division). The game should have two modes, practice and challenge:  **Practice Mode:**   * Users should keep answering questions until they select to stop. * When a user answers a question, they either:   + Are told it’s correct   + Are told their wrong, and provided with the correct answer * When the user leaves practice mode, they are given a summary, detailing:   + How many correct answers they got from hom many questions asked   + Each question, what they answered and what the correct answer was   **Challenge Mode:**   * Users answer questions for 30 seconds * They user gets no immediate feedback * At the end of the quiz they are given a summary, detailing:   + How many correct answers they got from hom many questions asked   + Each question, what they answered and what the correct answer was |
| **Option 2 – Who wants to be a Millionaire**  Create a computer game to replicate the T.V. game show 'Who wants to be a Millionaire' .  In the show, the contestant has to answer up to fifteen general knowledge questions in return for cash prizes. The contestant is shown each question together with the four multiple choice answers in advance of deciding whether he/she wishes to continue to play. Prior to answering any question a contestant may withdraw from the competition and retain all cash so far won. If a question is incorrectly answered the contestant loses the money for which he/she is currently playing but will retain the winnings if he/she has reached one of the saftey levels ($1000, $32000). The contestant is entitled to use each of three 'Lifelines' on one occasion only which may be used either individually in respect of any question or in respect of the same question. The three Lifelines are : 'Phone a friend', 'Ask the studio audience', or '50:50'. In '50:50', two of the four multiple choice answers are eliminated. In 'Ask the studio audience', the contestent is shown the percentage of the studio audience who have voted for each of the four possible answers. |
| **Option 3 – Hangman**  You are required to design a program that will play the game of Hangman with the user. Hangman is a classic word game in which you guess a secret word letter by letter. Initially, you know only the number of letters in the word.  The user guesses a letter. If they guess a letter that is in the secret word, the letter will be substituted everywhere it appears in the word. If they guess a letter that is not in the word, another part will be added to the hanged man (head, arms, torso, etc). If the user guesses all the letters in the word, they win! Once all the parts of the hanged man have been drawn, the man is hanged, and the user loses. |

**SENIOR INFORMATION PROCESSING & TECHNOLOGY**

**Semester 1 / 3, 2016**

**Minor Project Assessment Criteria**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student name:** | |  | | | |  | |  | |
| **STANDARD ACHIEVED** | **KNOWLEDGE & APPLICATION** | |  | **ANALYSIS & SYNTHESIS** |  | | **EVALUATION & COMMUNICATION** | |  |

|  |
| --- |
| **Comments:** |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KNOWLEDGE & APPLICATION** | **A** | **B** | **C** | **D** | **E** |
| ***Planning***  External design documentation – all sections are included | A **detailed** **description** and **explanation** of the planned game is presented | A **description** and **explanation** of the planned game is presented | A **description** of the planned game is presented | A **statement** about the planned game is presented | I**solated facts** about the game are presented |
| ***Design***   * Screen designs * List Data types and functions * Algorithm | **detailed** and **effective** application of set processes to design the game’s algorithm | **effective** application of set processes to design the game’s algorithm | application of set processes to design the game’s algorithm | **elements** of set processes to design the game’s algorithm | elements of set processes to design **parts** of the game’s algorithm |
| ***Implementation***  Screen Designs  Correct use of:   * Data types * Files * Arrays * Sequence * Selection * Iteration * Modular programming | **detailed** and **effective** application of set processes to implement the game | **effective** application of set processes to implement the game | application of set processes to implement the game | **elements** of set processes to partially implement the game | **elements** of set processes to implement an **incomplete** game |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ANALYSIS & SYNTHESIS** | **A** | **B** | **C** | **D** | **E** |
| ***Planning***  Problem definition  System objectives | **detailed** interpretation and analysis of problems and situations | **interpretation** and analysis of problems and situations | **analysis** of problems **and** situations | **identification and classification** of problems or situations | restated problems or situations |
| ***Design***  Efficient and effective use of the algorithm | designed an **effective and efficient** solution in constructing the game | designed a solution in constructing the game | designed a **partial** solution in constructing the game | designed **elements of** a solution in constructing the game | designed **superficial elements of** a solution in constructing the game |
| ***Implementation***  Efficient use of:   * Data types * Files * Arrays * Sequence * Selection * iteration * Modular programming | developed an **effective and efficient** solution in constructing the game | developed a solution in constructing the game | developed a **partial** solution in constructing the game | developed **elements of** a solution in constructing the game | developed **superficial** **elements of** a solution in constructing the game |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EVALUATION & COMMUNICATION** | **A** | **B** | **C** | **D** | **E** |
| ***Communication Skills***  Python Coding  • Internal code documentation  • Appropriate and consistent indentation  • Choice of suitable identifiers for variables and components | **comprehensive** construction of documentation and **fluent** presentation of information using **suitable** communication conventions to convey meaning **appropriate to the context** | **effective** construction of documentation and **effective** presentation of information using **suitable** communication conventions to convey meaning **appropriate to the context** | **construction of documentation** and presentation of information using communication conventions **to convey meaning** | presentation of information **using elements of communication conventions** | presentation of information |
| ***Communication Skills***  User Manual   * Ability to present your project making points succinctly, with fluency, clarity and using appropriate technical language. * User manual ease of understanding and competence. | **comprehensive** construction of documentation and **fluent** presentation of information using **suitable** communication conventions to convey meaning **appropriate to the context** | **effective** construction of documentation and **effective** presentation of information using **suitable** communication conventions to convey meaning **appropriate to the context** | **construction of documentation** and presentation of information using communication conventions **to convey meaning** | presentation of information **using elements of communication conventions** | presentation of information |
| ***HCI***  *System user friendliness – ability to create a system considering user perspective at all times.*  *User interface* | **fluent** presentation of information using suitable communication conventions to convey meaning appropriate to the context. | **effective** presentation of information using **suitable** communication conventions to convey meaning **appropriate to the context**. | presentation of information using communication conventions **to convey meaning.** | presentation of information **using elements of communication conventions**. | presentation of some information |
| ***Testing and Evaluation***  Evaluation with accompanying evidence.  Evaluation of the process  Evaluation of the product with direct reference to the program objectives | **comprehensive** testing of processes **and** solutions, application of **self-determined** and **prescribed** criteria, reasoning **and** evidence to draw conclusions and make **supported** recommendations | **reliable** testing of processes **and** solutions, application of **prescribed** criteria, reasoning **and** evidence to draw conclusions and make **supported** recommendations | testing of processes **or** solutions, application of **prescribed** criteria, reasoning **or** evidence to draw conclusions and make recommendations | **elements** of testing of processes **or** solutions to draw inferences | **elements** of testing |