## Lesson plan 1:

## Exam technique for extended response questions

|  |  |
| --- | --- |
| **Objectives**   * To be able to produce a correctly structured extended response answer to an examination question. * To understand what a level based mark scheme is and how they are applied to extended response answers. * To increase knowledge and understanding of the topic being assessed. | **Resources**  Question 3(a) Sample Assessment Material Paper 1: Principles of Computer Science.  Print a copy of the question for each student.  Have the mark scheme available electronically.  Use ‘pose, pause, pounce, bounce’ technique <https://www.youtube.com/watch?v=i6KCrLNCNT0> |
| **Transferable skills**  **Intrapersonal skills:**   * Continuous learning (Expanding skills and skill-sets through learning and increasing knowledge) * Intellectual interest and curiosity (Willing to tackle challenging problems; researching appropriate solutions and seeking to widen knowledge and increaseunderstanding. * Initiative (Using computational skills to develop own understanding and problem solving abilities.) * Integrity (Taking ownership of their own work and responding to challenges.) * Self-monitoring/self-evaluation/self-reinforcement (Planning and reviewing own work as a matter of course.)   **Interpersonal skills**  Self-presentation (Presenting outcomes of a group or  individual task to the whole class.   * Communication (Able to communicate ideas to peers and teachers and to discuss the logic of algorithms and code (verbally or written). * Collaboration (Working with peers on shared tasks; giving feedback on peers on problem solving and other tasks. * Teamwork (Working with peers to solve problems and create programs. * Co-operation (Share ideas with peers and supports peers who are finding tasks difficult.) * Interpersonal skills (Giving feedback to peers that is appropriate and delivered in a way that encourages them.)   **Cognitive skills**   * Problem solving (Solving problems is the core of computer science. Problem solving skills include breaking a large problem into a number of smaller ones, recognizing how problems relate to ones that have been solved, setting aside details of a problem that are less important, and identifying and refining the steps needed to reach a solution. * Analysis (Organising ideas and thinking both creatively and logically. * Reasoning/argumentation (Piecing together information, usually to recommend or suggest further action. * Decision making (Judging multiple options or alternatives, in order to select one so as to best fulfil requirements/needs. * Adaptive learning (Using skills, knowledge and understanding to respond the new and unfamiliar challenges. * Executive function (Analysing a situation, planning and taking action, maintaining attention, and adjusting actions as needed to complete a task. * Creativity (Using efficient/effective strategies to create a solution/solve a problem |

|  |
| --- |
| **Preparation** |
| For this lesson students will need knowledge and understanding of the topic being assessed. This could be imparted using one of these methods:   * In a previous lesson * In guided research completed as homework in advance of the class (see example at end of this lesson plan) * Using flipped learning, where the student has acquired the knowledge and understanding themselves prior to the lesson. This assumes videos/podcasts have been created previously.   The topic in this lesson is secondary storage, referenced in the specification at:  **4.2.5** Understand how data is stored on physical devices (magnetic, optical, solid state).  **4.2.6** Understand the concept of storing data in the ‘cloud’ and other contemporary secondary storage. |

|  |
| --- |
| **Starter** |
| Divide class into four groups:   * Give each group one topic from magnetic, optical, solid state and cloud storage. * Ask each group to put together a short verbal explanation (one sentence) on their topic. * Each group should then deliver the explanation   After each explanation ask the class as a whole to suggest scenarios where their topic could be the most appropriate.  Continuous learning, integrity, communication, teamwork, co-operation, reasoning/argumentation. |
| **Main lesson** |
| Issue Q3(a) from the SAM Paper 1.  **Analyse what the question is asking:**   * Give students a few minutes to read and discuss. * Ask what they think the question is asking for (POSE, PAUSE, POUNCE, BOUNCE).   + Challenge answers until agreement reached, e.g. ‘is that all, anything else?’   + Reinforce not yet answering the question. * Ask for any concerns about answering the question * Review these concerns by taking suggestions from other students * Summarise on the board   + Get agreement from all, question all   **Review technique for answering the question, explain that:**   * Marks are achieved by being able to justify their solution * The mark scheme gives ‘indicative content’ (preferred choice of content in this question content) which should be identified * Then marks are awarded on how well the answer is constructed – show levels based mark scheme at the top of page 31 in the mark scheme (don’t show indicative content). Review with the class.   **Students identify indicative content:**  Split students into pairs.  Ask them to   * Identify the secondary storage medium most suitable for the team and justify why it best meets their needs. * They should make notes covering what they have discussed.   Now join each pair with another (4 students in each group)   * The group should then discuss the notes they have made so far, and combine them. * Encourage students to review their initial work and review what peers have contributed, positively.   Now join each group of 4 with another (8 students in each group)   * The group should again discuss the notes they have made so far, and combine them. * Encourage students to review their initial work and review what peers have contributed, positively.   This group combining and sharing should continue until the class is a single group.   * Summarise the points on the board. * Show the indicative content from the mark scheme and compare with the points the class has identified. * Let the students challenge/agree with the indicative points. Note that any appropriate answer may be accepted.   **Answer the question:**   * Show the levels based mark scheme at the top page 31. Take questions from students. * Allow 10 minutes for the students to individually answer the question. * Show the indicative content on page 30. * Swap answer with a partner, and mark.   continuous learning, intellectual interest and curiosity, initiative, Self-monitoring/self-evaluation/self- reinforcement, communication, collaboration, teamwork, co-operation, interpersonal skills, problem solving, analysis, adaptive learning, executive function, creativity, |
| **Plenary** |
| Ask students to note down quickly (first impressions are important):   * One thing they have learned from the lesson * One thing they are still unsure about and don’t know yet.   Note down on board and use what they don’t know yet as starter for next lesson.  Continuous learning, integrity, communication, co-operation, analysis reasoning/argumentation, decision making, executive function. |

This task can be given to students to prepare in advance for the lesson.

|  |
| --- |
| Research task: Secondary storage devices In this task you are going to research certain topics. You can use the world wide web and/or textbook resources.  You must use **at least** 2 resources for each topic, and reference these resources.  You must then **use your own words** to complete the task.   1. Explain how data is stored on: 2. Magnetic devices 3. Optical devices 4. Solid state devices 5. Explain what ‘the cloud’ is and how data is stored in ‘the cloud’. |

## Lesson plan 2:

## Improving debugging skills

|  |  |
| --- | --- |
| **Objectives**   * To improve debugging skills whilst programming. * To increase confidence and ability in debugging as preparation for the practical examination. | **Resources**  Programming language used by the students. |
| **Transferable skills**  **Intrapersonal skills**   * Adaptability (Persisting in the face of difficulties, such as when writing code or designing an algorithm.) * Continuous learning (Expanding skills and skill-sets through learning and increasing knowledge) * Intellectual interest and curiosity (Willing to tackle challenging problems; researching appropriate solutions and seeking to widen knowledge and increaseunderstanding. * Initiative (Using computational skills to develop own understanding and problem solving abilities.) * Responsibility (Taking responsibility for finding and correcting errors in coding and algorithms. * Perseverance (Seeking to remove all errors in code and algorithms using testing and other tools.   **Interpersonal skills**   * Communication (Able to communicate ideas to peers and teachers and to discuss the logic of algorithms and code (verbally or written). * Collaboration (Working with peers on shared tasks; giving feedback on peers on problem solving and other tasks. * Teamwork (Working with peers to solve problems and create programs. * Co-operation (Share ideas with peers and supports peers who are finding tasks difficult.) * Interpersonal skills (Giving feedback to peers that is appropriate and delivered in a way that encourages them.)   **Cognitive skills**   * Critical thinking (Clarifying thoughts sufficiently so that they can be expressed in a form that a computer can carry out Analyzing and identifying possibilities and strategies to meet a required process and outcomes. Identifying the outcome of a computational process. * Problem solving (Solving problems is the core of computer science. Problem solving skills include breaking a large problem into a number of smaller ones, recognizing how problems relate to ones that have been solved, setting aside details of a problem that are less important, and identifying and refining the steps needed to reach a solution. * Analysis (Organising ideas and thinking both creatively and logically. * Reasoning/argumentation (Piecing together information, usually to recommend or suggest further action. * Decision making (Judging multiple options or alternatives, in order to select one so as to best fulfil requirements/needs. * Adaptive learning (Using skills, knowledge and understanding to respond the new and unfamiliar challenges. |

|  |  |
| --- | --- |
| **Background** | |
| This lesson is designed to reinforce knowledge and understanding of debugging whilst programming by using transferable skills and can be used in any programming lesson.  The examples here use the introduction of bubble sort which is then programmed using arrays (lists in Python).  Teachers are of course free to use their own resources for bubble sort theory but the examples here utilise the Pearson iGCSE Computer Science Teaching and Learning Materials [here](https://qualifications.pearson.com/en/qualifications/edexcel-international-gcses-and-edexcel-certificates/international-gcse-computer-science-2017.coursematerials.html#filterQuery=Pearson-UK:Category%2FTeaching-and-learning-materials), specifically Scheme of Work 2017 Year 1 Term 3, Week 11 Lesson 2. | |
| **Preparation** | |
| If required, obtain resources from Scheme of Work 2017 Year 1 Term 3, Week 11 Lesson 2.  Prepare a pseudo-code algorithm of a bubble sort, or use the example later in this lesson.  Prepare buddy pairing cards with numbers and names. Each student has a different number and the buddy they call on will be the next in the sequence. For example, student 5 will have as a debugging buddy student 6, who will have as debugging buddy student 7.  For example: | |
| 5 | Waheed Khan |
| 6 | Jane Smith |
| 7 | Chen Wong |
| The last student will have the first student as a debugging buddy. | |
| **Starter** | |
| Divide class into small groups:   * Explain that they will be introduced to a new concept (bubble sort) which they will then program using arrays/Python lists * The group should review their understanding and collectively note down one thing they are confident about and one thing they are not yet confident about. * Review the groups’ understanding and explain that they will be learning how to improve their debugging skills which will increase their confidence and understanding. Remind them that this will assist them when they are undertaking the practical examination.   Continuous learning, communication, teamwork, reasoning/argumentation. | |
| **Main lesson** | |
| **Teaching points** Theory  * Spend about 20 minutes explaining bubble sort.  Set up task:  * Give each student a pseudo-code algorithm of a bubble sort. * Give each student a copy of the debugging buddy flowchart. * Students are assigned their debugging buddy. * Give a list of numbers to be sorted, e.g.:   + 16, 8, 4, 25, 10, 29, 12 * Ask students to start creating the program.   Explain that they are going to use debugging buddies to help them debug any problems as they create the bubble sort program.  **Note:** it is not essential that the program is completed in this lesson, it can be continued in the next one.  **Rules of engagement:**   * The buddy system is a partnership; students need to work together. * Students should not call on their friends to help instead of their buddy. * All students should verbalise what they are doing as they go (i.e. buddies should not just jump in and fix it!) * When an explanation is given, the understanding of the other person should be checked.   The teacher’s role is to monitor but **not** to assist in debugging.  It may be helpful to suggest different ways of analysing the problem (e.g. explain verbally, trace through (manually or using the IDE debugging tools), using standard conventions like naming variable appropriately etc.).  Encourage all students to participate – this does not need strong programming skills.  Adaptability, continuous learning, intellectual interest and curiosity, initiative, responsibility, perseverance, communication, collaboration, teamwork, interpersonal skills, critical thinking, problem solving, analysis. | |
| **Plenary** | |
| Divide students into their groups again.   * Ask each group if they have been able to improve their confidence and understanding * From each group, take one positive and one thing that could be improved about the debugging buddy method.   Continuous learning, communication, teamwork, reasoning/argumentation. | |

START

N

N

N

N

Y

Y

Y

Y

**Debugging Buddy Flowchart**

FINISH

Debugged?

Call for next buddy to join them

Explain problem

Explain what has been done to debug

Debugged?

Buddy works with student to debug

Debugged?

Call for buddy

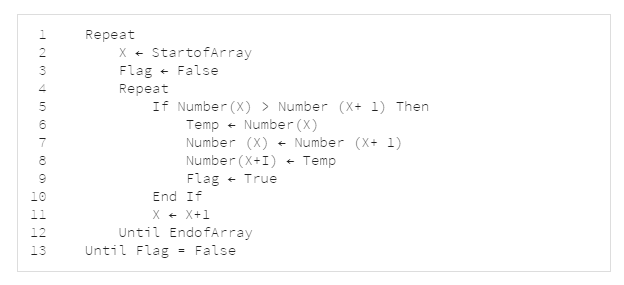
Explain problem

Explain what has been done to debug

Debugged?

Try to debug problem

## Bubble sort algorithm



Ref: <http://www.pythonschool.net/data-structures-algorithms/bubble-sort/>

## Lesson plan 3:

## Understand how encryption algorithms work

|  |  |
| --- | --- |
| **Objective**   * To understand how the encryption algorithms in the specification (Pigpen cipher, Caesar cipher, Vigenére cipher, Rail Fence cipher) work. | **Resources**  Caesar cipher wheel template here:  <https://fprint.net/wp-content/uploads/printable-cipher-wheel.pdf>  Vigenére cipher template here:  <https://commons.wikimedia.org/wiki/File:Vigenere-square.png>  Links to algorithms:  Pigpen cipher:  <http://www.simonsingh.net/The_Black_Chamber/pigpen.html>  <https://www.youtube.com/watch?v=9MlW6Kpp8Xk>  Caesar cipher:  <https://inventwithpython.com/hacking/chapter1.html>  <http://www.simonsingh.net/The_Black_Chamber/caesar.html>  Vigenére cipher:  <http://www.simonsingh.net/The_Black_Chamber/vigenere_cipher.html>  <https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher>  Rail Fence cipher:  <http://www.braingle.com/brainteasers/codes/railfence.php>  <http://practicalcryptography.com/ciphers/rail-fence-cipher/> |
| **Transferable skills**  **Intrapersonal skills:**   * Continuous learning (Expanding skills and skill-sets through learning and increasing knowledge) * Intellectual interest and curiosity (Willing to tackle challenging problems; researching appropriate solutions and seeking to widen knowledge and increaseunderstanding. * Initiative (Using computational skills to develop own understanding and problem solving abilities.) * Integrity (Taking ownership of their own work and responding to challenges.)   **Interpersonal skills**   * Communication (Able to communicate ideas to peers and teachers and to discuss the logic of algorithms and code (verbally or written). * Collaboration (Working with peers on shared tasks; giving feedback on peers on problem solving and other tasks. * Teamwork (Working with peers to solve problems and create programs. * Co-operation (Share ideas with peers and supports peers who are finding tasks difficult.) * Interpersonal skills (Giving feedback to peers that is appropriate and delivered in a way that encourages them.)   **Cognitive skills**   * Problem solving (Solving problems is the core of computer science. Problem solving skills include breaking a large problem into a number of smaller ones, recognizing how problems relate to ones that have been solved, setting aside details of a problem that are less important, and identifying and refining the steps needed to reach a solution. * Analysis (Organising ideas and thinking both creatively and logically. * Reasoning/argumentation (Piecing together information, usually to recommend or suggest further action. * Decision making (Judging multiple options or alternatives, in order to select one so as to best fulfil requirements/needs. * Adaptive learning (Using skills, knowledge and understanding to respond the new and unfamiliar challenges. * Executive function (Analysing a situation, planning and taking action, maintaining attention, and adjusting actions as needed to complete a task * Creativity (Using efficient/effective strategies to create a solution/solve a problem |

|  |
| --- |
| **Preparation** |
| The topic in this lesson is referenced in the specification at:  **3.4.1** Understand the need for data encryption.  **3.4.2** Understand how encryption algorithms work (Pigpen cipher, Caesar cipher, Vigenére cipher, Rail Fence cipher).  Teachers should familiarise themselves with how these algorithms work using the hyperlinks in the resources column.  Set up the classroom:   * Arrange the classroom so that there are 4 different ‘stations’ with a central group of tables and chairs around them. With larger groups create a multiple of 4. * Print off the cipher challenges (see later in the lesson). There should be one for each student on each table, plus a few spares. * Have plenty of spare paper on each table for students to use when working out. * For the Caesar cipher challenge, make enough wheels for each student at each station. * For the Vigenére cipher challenge, print enough copies of the Vigenére square for each student at each station. * Have a document folder ready for each group to collect their workings and algorithms at each station (this is to avoid leaving work at stations which can be utilized by others; individual work will be distributed next lesson). |

|  |
| --- |
| **Starter** |
| Introduce the topic of the lesson – to understand the need for data encryption and how 4 algorithms work.  Explain the term ‘encryption’. One definition is:  *The translation of*[*data*](http://www.webopedia.com/TERM/D/data.html)*into a secret code. Encryption is the most effective way to achieve data*[*security*](http://www.webopedia.com/TERM/S/security.html)*. To*[*read*](http://www.webopedia.com/TERM/R/read.html)*an encrypted*[*file*](http://www.webopedia.com/TERM/F/file.html)*, you must have access to a secret*[*key*](http://www.webopedia.com/TERM/K/key.html)*or*[*password*](http://www.webopedia.com/TERM/P/password.html)*that enables you to*[*decrypt*](http://www.webopedia.com/TERM/D/decryption.html)*it. Unencrypted data is called*[*plain text*](http://www.webopedia.com/TERM/P/plain_text.html)*; encrypted data is referred to as*[*cipher text*](http://www.webopedia.com/TERM/C/cipher_text.html)*.* <http://www.webopedia.com/TERM/E/encryption.html>  Ask students to work in pairs and think of some ideas of where encryption would be needed to protect data.  Take suggestions from the class and note on board.  Continuous learning, integrity, communication, teamwork, co-operation, reasoning/argumentation. |
| **Main lesson** |
| Start the main part of the lesson by displaying a list of the 4 different encryption algorithms students need to understand on the board. These are:   * Pigpen cipher * Caesar cipher * Vigenére cipher * Rail Fence cipher   Explain the difference between **ciphertext** and **plaintext**.   * Ciphertext is encrypted text * Plaintext is the original, or decrypted text   **Explain what is going to happen in this lesson**:   * Students will be divided into groups * They will spend a fixed amount of time at each station learning about a cipher by:   + Completing the challenges at that station   + Creating an algorithm as a group (using the method of their choice, i.e. flowchart, pseudocode, written description, program code).   **Start the task**:   * Split the class into 4 (or a multiple of 4) groups * Assign 1 student to collect all the group’s workings and algorithms from each station and keep safe in the document holder. * Ask students to write their names on any work they complete. * Tell the groups they will have a set time at each station (e.g. 10 minutes) * GO!   continuous learning, intellectual interest and curiosity, initiative, communication, collaboration, teamwork, co-operation, interpersonal skills, problem solving, analysis, adaptive learning, executive function, creativity, |
| **Plenary** |
| Ask students to vote for the algorithm that they think is the most **effective**. Tally the marks.  Now ask students to vote for the algorithm that they think is the most **ineffective**. Tally the marks.  Ask individual students randomly to justify their choice.  **Note**: Collect document folders and distribute notes to students at the next lesson.  Continuous learning, integrity, communication, co-operation, analysis, reasoning/argumentation, decision making, executive function. |

|  |
| --- |
| Pigpen Cipher – how it works: The Pigpen Cipher is a substitution cipher which substitutes each letter with a symbol. The symbol is the shape for each letter in a fixed grid.    A will be substituted by  K will be substituted by **.** Pigpen Cipher Challenges: Decipher these ciphertext examples into plaintext:    1)    2)  Now encrypt these examples using the pigpen cipher:  3) IT WAS A DARK AND STORMY NIGHT  4) BURN THIS PAPER AFTER YOU HAVE READ IT Caesar Cipher – how it works: This is a substitution cipher, where letters are substituted for each other based on a key. The key is the number of shifts from the plaintext to the ciphertext and the Caesar Cipher uses values between 0 and 25, for each letter of the alphabet. ‘A’ has the value of ‘0’.  A Caesar Cipher wheel is a useful tool since it makes it easy to go back to the start of the alphabet. Put the inner wheel with the key value under the outer wheel ‘A’.  It is then easy to read off the substitution values and encrypt or decrypt the message. Caesar Cipher Challenges: Decipher these ciphertext examples into plaintext:   1. FQNW MRM CQN BDW UJBC BQRWN (Key is 9) 2. JXU IFO XQT JE XYTU YD JXU RKIXUI (Key is 16)   Now encrypt these examples using the Caesar cipher:   1. It will not be long until the summer vacation   Choose your own key value.   1. The army sent a message   For this example, shift right using a key of 8, then left using a key of 3. |
| Vigenére cipher – how it works: This cipher uses 26 alphabets in a square, each one shifted by one letter. Any of the substation alphabets can be used, but is reliant on a keyword.  **Example:**   * The plaintext is ‘test this square’. Remove the spaces. * The keyword is ‘once’. * The keyword is then written repeatedly above each letter of the plaintext. * Then you must look at the column of the plaintext and the row of the keyword, and find the letter of the intersection. * The result is the encrypted text.   ONCEONCEONCEON (repeated keyword, use row)  TESTTHISSQUARE (plaintext, use column)  HRUXHUKEGDWEFR (encrypted text, find intersection)   Vigenére cipher challenges: Decrypt the following ciphertext examples into plaintext using the Vigenére cipher:   1. KVBKLLJSQV (keyword RED) 2. LJQWVUEUNMZRTGARIIE   Now make up 2 plaintext examples of your own, choose a keyword, and encrypt:  Keyword:  Plaintext:  Ciphertext:  Keyword:  Plaintext:  Ciphertext: |
| Rail Fence Cipher – how it works: This is a transposition cipher. First decide on the number of ‘rails’ the cipher will use. These will be the number of letters in each column. Then the letters are written down, snaking from column to column in order. To create the ciphertext now write all the letters down from the rows one after the other.  To decrypt you have to know the number of rails and the number of rows used. Split the text into equal rows (as far as possible) then write each letter down the rails. You may need to decide where spaces should be added.  **Example:**  **Plaintext:**  IT WAS A DARK AND STORMY NIGHT  **Rail fence:**  I A K D O H  T S D R N T  A A S M I  Note: when answering make sure the number of spaces is obvious  W A R N T Y G  **Ciphertext:**  IA\_KDO\_HTSD\_\_RNT\_\_AASMI\_WARNTYG Rail Fence Cipher Challenges: Decrypt the following ciphertext examples into plaintext using the rail Fence cipher:   1. I\_M\_E\_Y\_\_GAS\_AARTP (using 3 rails and 3 rows) 2. O\_NT\_L\_\_YNU\_IIAFA\_CPAMNNAW\_EO\_E\_DRA (using 4 rails and 4 rows)   Now encrypt the following plain text using Rail Fence. You decide on the number of rails, and note this down with the final number of rows.   1. When is my examination 2. I hope a get a very good mark in my examination |