N Scheme of Work 2020-21 Subject: Computing Unit 1 Principles of Computer Science						
Year Group: Year 12 **Specification: BTEC Computing – Unit 1 – this is to run co-currently as Unit 2 (students have 5 hours a week 2 hours for Unit 1 and 2 hours for Unit 2 and 1 hour self-directed learning)**						
Lesson No	**Topic & Objectives**	**Big Question – What will students learn?**	**Key Activities & Specialist Terminology (Do Now Task / Starter/Tasks/Plenary**	**Planned Assessment**	**Homework or flipped learning resources** **DODDLE resources**	**Lit Num SMSC Codes**
1 & 2	**A1 Decomposition**	**What is decomposition?**	• First lesson a baseline test should be completed. • **Lead-in:** Introduce the concept of computational thinking (CT) and stages of CT. Explain that CT will be applied throughout this and other units. • **Small group/paired activity:** Give learners a scenario in which they are asked to reverse-engineer a 'clone' of a simple computer game (eg Space Invaders or Tetris®). Learners must identify the distinct steps of the problem. **Link for resources:** There are different scenarios for you to pick for the students to do in the following link https://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/ • **Tutor-led discussion:** Learners feed their thoughts back to the whole group. • **Plenary:** Using what they have learned from the discussion, learners complete work on the scenario activity.	Baseline test	**Expansion – independent learning activity:** Students to use the following link to look for articles that illustrate decomposition – then they are to write a small paragraph detailing the main points https://teachinglondoncomputing.org/resources/developing-computational-thinking/decomposition/ Exercises on problem solving (CT) from PG Online resources	Lit Social So8 C3 Sp2 Sp5

3 & 4	**A2 Pattern recognition**	**What is the pattern? Read and understanding code**	• **Lead-in:** Recap the concept of decomposition with a Q&A session. Explain that in this lesson they will continue to explore CT. • **Small group/paired activity**: In the same groups as the previous lesson, learners expand their analysis of the example game (eg identify patterns and common elements or features). Introduce some programming terms and ask learners to identify variables in the example game (eg 'lives'). • **Tutor-led discussion:** Ask learners to contribute their thoughts to the discussion. • **Plenary:** Using what they have learned from the discussion, learners complete work on the scenario activity in the following link https://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/ **(Teacher's note:** the above link contains lots of different activities for you to choose the ones suitable for your learners)	Exam questions on Problem Solving (CT) – this to be complete on their own – use peer marking (pg 39 PG Online Book)	Teachers to pick an activity from the following link that gets students to identify elements of code https://teachinglondoncomputing.org/resources/inspiring-unplugged-classroom-activities/ Computational Thinking - Problem Solving Worksheet 1 + Homework sheet 1	Lit Social So8 C3 Sp2 Sp5
5 & 6	**A3 Pattern generalisation and abstraction**	**How do you identify the variables?**	• **Small group/paired activity:** In the same groups as the previous lesson, learners expand their analysis of the example game (eg identify patterns and common elements or features). Introduce some programming terms and ask learners to identify variables in the example game (eg 'lives'). • **Tutor-led discussion:** Ask learners to contribute their thoughts to the discussion.	Exam questions – PG online – Constants and Variables (pg6-7)	Activities from BTEC revision guide pg 1-6 Programming – Worksheet and Homework sheet 1	Lit Social So8 C3 Sp2 Sp5

			• **Plenary:** Using what they have learned from the discussion, learners complete work on the scenario activity.			
7 & 8	**A4 Algorithm design**	**How to write an algorithm – flowchart introduction.**	• **Lead-in:** Recap the concepts of 'decomposition', 'pattern recognition' and 'pattern generalisation and abstraction' through Q&A session. Explain that over the next two lessons learners will explore the first stages of algorithm design. • **Small group or paired activity:** Using the same groups as previous lessons, learners consider the 'algorithms' for different parts of the game (eg controlling the ship, firing or invader movement). Highlight that in the early design stages of a software solution the 'algorithms' only need to be step-by-step instructions, using everyday language, and do not need to be presented in pseudocode or full programming code. • **Group activity (peer review and feedback):** At different stages during the two lessons, ask groups to share their work with other small groups. Each group should give feedback about the algorithms produced. Feedback should focus on completeness and accuracy. • **Plenaries:** Give groups different-coloured pens to help them annotate work when giving feedback, and so they can annotate changes they have made as a result of feedback. • **Extension – independent learning activity:** If time and resources allow, give learners a chance to implement some of their algorithms into a working	Deep Assessment in Purple books on topics learnt over the last couple of weeks – with exam style question (taken from Sample Papers – BTEC online)	Exercise on Algorithm Design BTEC Revision Guide Pg 8 Structured Programming – Worksheet and Homework sheet 2	Lit Social So8 C3 Sp2 Sp5

			system, eg the 'invader movement' algorithm using Scratch. Learners could use pre-made sprites and apply their algorithm. They should check their algorithm by running the 'code' to see if it performs as they would expect.			
9 & 10	**B1 Structured English (pseudocode)** • Interpreting pseudocode. • Producing pseudocode. **B2 Flowcharts using standard symbols**	**How do you read and write pseudocode?**	• **Tutor presentation:** Introduce the concept of pseudocode when designing algorithms and planning computer programs. Explain why pseudocode is used as a planning and design tool (ie to identify and plan the correct sequence of tasks/processes for a computer program). Introduce the 'basic operations' listed in the specification. Explain that operations, key commands and common programming functions are typically written in upper case within the pseudocode. See the following video link: Pseudocode Tutorial (6.5 minutes) www.youtube.com/watch?v=rSz7549WSjY • **Independent learning activity:** Give learners simple scenarios and flow charts or components of a computer program that they must present as pseudocode. Focus on the basic operations and avoid, if possible, decisions and repetition. However, depending on learners' ability and their programming experience, you may decide to introduce concepts from later lessons. *Note: try to introduce a different scenario from previous lessons. It is important that learners can apply knowledge and demonstrate understanding in a range of contexts.*	Clearly written pseudocode that contains a clear result	BTEC Revision Guide pg 9 Pg 12 Exercises in PG Online Book	Lit Social So8 C3 Sp2 Sp5

			They should be able to work from scenarios and/or flow charts. Sequence-based scenarios could include: inputting a number, squaring it and outputting the results temperature conversion between Fahrenheit and Celsius calculating the area of a circle. • **Plenary:** Ask learners to discuss/compare their solutions in small groups. Ask one group to share their solutions. As a class, discuss the merits of each solution (try to identify a group that has two varying but still valid solutions). Highlight that, depending on the scenarios, there may be different ways of presenting solutions which are still valid, just as there are when creating a programmed solution.			
11 & 12	**B1 Structured English (pseudocode)** • Interpreting pseudocode. • Producing pseudocode. **B2 Flowcharts using standard symbols**	**How to read and write pseudocode to solve problems in previous coded work.**	• **Lead-in:** Recap learners' understanding by asking them to produce an algorithm for a simple scenario using 'basic operations'. • **Tutor presentation:** Introduce the 'decisions' listed in the specification. Reiterate that operations, key commands and common programming functions are typically written in upper case within the pseudocode. Use Q&A with some example logical problems to establish learners' understanding. Explain that indentation can be used to show hierarchy of tasks. • **Independent learning activity:** Give learners some simple scenarios, and where appropriate supporting flow charts, or components of a computer program that require the use of	Deep Assessment on the computer of what code they have learnt over the last couple of weeks. Students to screen shot their code and print for marking and grading – Exam questions from Unit 1 Papers used to assess students fully.	Exercises Pg 42-50 PG online book	Lit Social So8 C3 Sp2 Sp5

			'decisions'. They should write pseudocode to represent the solutions. Selection-based scenarios could include: • cinema tickets (Adult, Child, Senior and Student rates) • calculating the cost of an invoice, with free delivery for orders of £100. • **Tutor-led class discussion:** Go through the solutions as a class. Select examples that highlight correct solutions and also some that highlight common misconceptions. • **Plenary:** Give learners time to correct or develop their pseudocode algorithms. Or ask them to annotate their pseudocode to explain its function.			
13 & 14	**B1 Structured English (pseudocode)** • Interpreting pseudocode. • Developing pseudocode. Producing pseudocode.	**What is nested logic and repetition?**	• **Lead-in:** Recap learners' understanding by asking them to produce an algorithm for a simple scenario using 'decisions'. • **Tutor presentation:** Introduce the concepts of 'nested logic' and repetition. Explain that in computer programs 'decisions' are often not stand-alone and conditions are regularly chained or combined with repetition to reach an outcome. • **Independent learning activity:** Give learners a series of varying scenarios and/or problems that require 'nested logic' and/or repetition. They should write pseudocode to represent the solutions. *Note: you could give a wider range of possible scenarios by introducing some*	Written pseudocode that clearly represents a solution	Exercises on Pg 10 – 11 BTEC Revision Guide	Lit Social So8 C3 Sp2 Sp5

			of the arithmetic operations listed in topic C2. Iteration-based scenarios could include the following. • Guessing game - the computer randomly generates a number between 1 and 30 and the player has to guess the number. For each guess the computer responds with higher or lower and the player has a limited number of guesses. • Palindrome problem - inputting a word, reversing it, comparing the results and declaring whether or not it was a palindrome. • **Plenary – paired activity:** Ask learners to compare and contrast their solutions. Support them to identify any errors in the logic and allow time for them to correct their solutions.			
October Half Term						
15 & 16	**B1 Structured English (pseudocode)** • Interpreting pseudocode. • Producing pseudocode. **B2 Flowcharts using standard symbols**	**What improvements can be made to the written pseudocode?**	• **Lead-in:** Organise learners so they can complete assessment tasks independently. **Individual activity:** Give learners a series of scenarios and flow charts. They should produce pseudocode that demonstrates their learning over the last few lessons. The activities should increase in demand as the assessment progresses. Give learners at least one scenario where they have to evaluate pseudocode and suggest improvements or write an improved version. *NB: try to structure and present activities in a similar way to the sample assessment material supporting the unit.* Problems could include:	Written pseudocode with improvements	Writing and interpreting algorithms worksheet 3.	Lit Social So8 C3 Sp2 Sp5

			• password problem (maximum three attempts) • setting up a password (must be a certain length and contain an upper case letter, a number and a symbol) • weights and measures convertor. Discuss the command verbs and how learners should structure their answers when presented with them. Definitions are in the specification.			
17 & 18	**C1 Handling data within a program** Defining and declaring constants and variables.	**Identifying what is the best data type to use in different cases.**	• **Lead-in:** Explain that over the next few lessons learners will explore the use of data and variables in computer programs. • **Tutor presentation:** Give learners a list of the data types in the specification. Identify the properties/characteristics of each data type and explain the difference between the primitive data types and the composite data types. • **Paired activity:** Give learners a series of examples of pseudocode containing a range of variables and example data. In pairs, learners should discuss the examples and identify which data types they would use in each case. • Float examples - currency problems such as conversion. • String examples - counting the occurrence of a letter in a string or reversing a string. • Integer examples - problems using counters (for example displaying a bar chart). • **Class discussion:** Discuss the decisions made during the activity.	Deep Assessment in Purple books on topics learnt over the last couple of weeks – with exam style question - Question 1a & b from Exam paper June 2017	Pg 12 & 13 BTEC Revision book	Lit Social So8 C3 Sp2 Sp5

			• **Individual activity:** Give learners an additional example scenario and specification for a computer program that would use different data types (eg a program that could handle personal information for employees of a company). Try to give learners data that could possibly be stored using different data types and ask them to justify their choices.			
19 & 20	**C1 Handling data within a program** Managing variables.	**How computers handle data and understand what is the role of local and global variables?**	• **Lead-in:** Explain that learners will extend their work on how computers handle data and look at the role of local and global variables. • **Tutor presentation:** Introduce the concept of local and global variables. Explain how using them may affect the way a program works and the data within the program. Learners should think of global variables as being used by a number of different functions/subroutines (passing values will be explored later). See video link: Scope of variables (7.5 minutes) in the following link www.youtube.com/watch?v=RlQQK-mYmZs • **Individual activity:** Give learners pseudocode examples containing local and global variables that are used in different ways. Ask them to show what happens when different scenarios and data are passed in to the code. Validation routines in particular as the original copy of the data should be held back and only a copy worked on inside a function. • **Plenary:** Discussion regarding the validation used, discuss how relevant they were or not.	Give learners a written exam-style question to help summarise their learning. – Use Sample Learner paper	Pg 13 & 14 BTEC Revision book	Lit Social So8 C3 Sp2 Sp5

21 & 22	**C1 Handling data within a program** Managing variables.	**How to name conventions?**	• **Lead-in:** Explain that learners will extend their work on local and global variables and also look at naming conventions. • **Tutor presentation:** Recap the last lesson and explain how naming conventions might work. Then present some examples using simple programming code (Python may be a good starting point). Give examples (in pseudocode and Python) that show inconsistent/confusing approaches to variable names. Explain how in pseudocode they can still be understood but when executed they may cause different outcomes. • **Paired activity:** Give examples of pseudocode that require debugging to each pair of learners. Ask learners to identify problems within the code (either limited to naming conventions or extended to global and local variables). Learners should rewrite parts of the given code to correct problems and improve efficiency. • **Plenary:** Re-organise learners so that each pair joins another group. Ask each group to compare and contrast their solutions.	Re-written code that corrects the errors	Pg 16 & 17 BTEC Revision book	Lit Social So8 C3 Sp2 Sp5
23 & 24	**C1 Handling data within a program** Managing variables.	**Enforcing the concepts learnt in the previous sessions.**	• **Lead-in:** Reinforce the concepts explored over the last two lessons and explain that learners will continue to develop this work. • **Individual activity:** Give learners examples of pseudocode with a number of different problems and contexts. Ask them to complete tasks including explaining the functions, describing how data is processed by the code, debugging errors and re-	Clear explanations of the functions used, in the programming and description of how the data is processed by the code. AP testing – mock exam – Jan 2018 – Paper 1 & 2	Pg 18 & 19 BTEC Revision book	Lit Social So8 C3 Sp2 Sp5

			writing sections to improve efficiency. At this stage learners can still be working in pseudocode and do not need to create functioning programs.			
25 & 26	**C2 Arithmetic operations** **C3 Built-in functions** Arithmetic functions.	**What are high-level programming languages?**	• **Lead-in:** Explain that learners will look at common built-in functions that are available in many high-level programming languages. • **Tutor presentation:** Introduce the concepts of arithmetic functions. Explain the purpose and use of the functions and how they may be used to add functionality to a program. At this stage, just discuss and use these functions at pseudocode level. Learners will explore the syntax and use of these within specific languages later. However, you may wish to give examples of the functions working within a program, without going into the code, in order to give context. • **Individual learning activity:** Give learners a series of problems. Use pseudocode that contains the arithmetic operations from the specification. Ensure problems include a range of contexts in which learners must apply understanding. In particular these problems should include the need to process using BODMAS - they should be multi-stage calculations including nested brackets. • **Small group activity:** Organise learners into groups to improve their solutions.	Successful pseudocode code that makes sense and contains arithmetic. Solving problems that contain BODMAS – Learners must show clear understanding of al the contexts used.	Pg 20 & 21 BTEC Revision book	Lit Social So8 C3 Sp2 Sp5

27 & 28	**C2 Arithmetic operations** **C3 Built-in functions** • Arithmetic functions. • String handling functions. • General functions.	**What functions are available in high-level programming languages?**	• **Lead-in:** Explain that learners will continue to look at common built-in functions available in many high-level programming languages. • **Tutor presentation:** Introduce the string handling functions. Explain their purpose and use, and how they may be used to add functionality to a program. Explain how general functions are used as a matter of course when developing solutions. Again, at this stage these functions only need to be discussed and used at pseudocode level. Learners will explore the syntax and use of these within specific languages later. • **Individual learning activity:** Give learners a series of problems. Use pseudocode that contains the string handling functions from the specification. Ensure problems include a wide range of contexts in which learners must apply understanding. Allow time to discuss solutions with individuals or as small groups. Problems could include: A palindrome phrase checker, for example 'A car, a man, a maraca.' – see link in the following link www.palindromelist.net/A-car-a-man-a-maraca/ The spaces and punctuation in this phrase need to be stripped and the spaces closed up before reversing and comparing the strings.	Give learners a problem that requires the development of pseudocode. Try to give them a problem that requires application of arithmetic functions and string handling functions. The task should be completed in exam conditions and marked by the tutor to gauge progress. Adjust the support given depending on the learners' progress.	Research what 'Hungarian Notation' means in the context of naming conventions. What are the advantages and disadvantages of using this approach to naming?	Lit Social So8 C3 Sp2 Sp5