BTEC Level 3 Computing for Creative Industries

Unit 01

Fundamentals of Computer Science

Table of Contents

Key terms

Complete this list as you study for Unit 01; you are expected to understand, be able to use, and correctly spell all of the key terms in this list.

Term	Meaning
Abstraction	
Algorithm	
Alphanumeric	
Arithmetic functions (round, truncate)	
Arithmetic operations (+, -, *, /, %)	
Array (single-, multi- dimensional)	
Binary search	
Boolean	
Branch	

Bubble sort	
Callback function	
Character	
Class	
Concatenation	
Condition	
Constants	
Constraints (validation)	
Control structure	
Count occurrence algorithm	
Data hiding	
Data type	

Decision	
Definite iteration	
Encapsulation	
Event	
Event driven programming	
Event loop	
Event handler	
FIFO	
Flowchart	
Floating point	
FOR	
Function	

Global variable	
Indefinite iteration	
Indentation	
Inheritance	
Input validation	
Insertion sort	
Instance	
Integer	
Iterative control	
LIFO	
Linear search	
List	

Local variable	
Main loop	
Object	
Object oriented programming	
Output	
Overloading	
Paradigm	
Platform	
Protocol	
Postcheck action	
Polymorphism	
Procedure	

Procedural programming	
Pseudocode	
Quick sort	
Repetition	
Records	
Relational operator	
Reusability	
Scope	
Service oriented processing	
Sub routine	
Sequence	
Statement	

Structure	
Serial search	
Sets	
Strings	
Time driven	
Translation	
Trigger function	
Variable	

Topics

Computational thinking

- Decomposition
- Pattern recognition
- Pattern generalisation and abstraction
- Algorithm design

Standard methods and techniques used to develop algorithms

- Structured English
- Flowcharts

Programming paradigms

- Handling data within a program
- Arithmetic operations
- Built-in functions
- Validating data
- Control structures
- Data structures
- Common/standard algorithms

Types of programming

- Procedural programming
- Object-oriented programming
- Event driven programming
- Coding for the web
- Translation

For further information about the contents of each section, please see the specification.

Exam-style questions

1. Tom would like to set up a website. He is planning on offering items made by local craftsmen for sale. He would like to have users sign up on his website so that he can keep them up to date when new items are made available.
   Tom has heard people talking about server-side and client-side code, and isn't sure what he will need to make his website work. Explain whether Tom will need either server-side or client-side code, or a combination, and why you believe this to be the case. [12]

2. Danni has been asked to design a feedback system for use in a shopping centre. It should allow users to quickly press a button to indicate how their shopping trip was. For example, whether they found everything they needed or not. What features of event driven programming would be ideally suited for such a project? [10]

3. Stuart wants to write a program which models a population of insects and predators within a small ecosystem. Each type of insect has different properties and attributes. What paradigm would be most appropriate for Stuart to use when writing his code, and how/why? [12]

4. Jasmine has been writing an application for her employer using assembly code. Her employer has suggested that it would be a good idea to migrate the existing code-base to a higher-level language such as C#. Discuss the advantages and disadvantages of doing this. [12]

5. John is concerned that the programming language he is using is not being actively developed any more. He feels that it would be a good idea to translate the code over to a newer language. Why might John feel this way? [10]

6. What security precautions would you recommend to a client who wishes to set up a website that allows visitors to record short memos, and why would these recommendations be worthwhile? [12]

How to approach written questions

ALWAYS remember the following points:

- **Answer questions as if you are explaining to a non-specialist**. For example, even on a short question such as “What is a statement? [3]”, you must make three points. If you were talking to a non-specialist, you might say:
    - A statement is a single line of code
    - That achieves something
    - Such as setting the value of a variable

- You are not going to have to write significant amounts of code. Therefore long answers will ask “how well do you know this topic?” **A question phrased as “Why would object oriented programming be appropriate to use in game design?” should be treated as the following two questions (and similarly for all other variants):**
    - What is object oriented design? **
    - Given some examples of how those features could be used when designing a game

  Alternatively, “Why would event driven programming be suitable for the proposed system?” means:
    - What is event driven programming? **
    - Given example of how those features apply to the proposed system

  ** **In all cases**, before you start writing the answer, having worked out what the question is actually asking you, write a list of keywords for that topic, and when explaining them, remember to always include:

    - The word
    - A definition
    - An example

Revision resources

Object oriented programming concepts

https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/object-oriented-programming

https://realpython.com/python3-object-oriented-programming/

Event driven programming

http://www.technologyuk.net/computing/software-development/software-design/event-driven-programming.shtml

https://dzone.com/articles/introduction-event-driven

Procedural programming

https://en.wikipedia.org/wiki/Procedural_programming

Standard algorithms

https://en.wikipedia.org/wiki/Binary_search_algorithm

https://www.tutorialspoint.com/data_structures_algorithms/linear_search_algorithm.htm

https://en.wikipedia.org/wiki/Bubble_sort

https://en.wikipedia.org/wiki/Quicksort

https://www.tutorialspoint.com/data_structures_algorithms/merge_sort_algorithm.htm

https://en.wikipedia.org/wiki/Insertion_sort

Web development

https://dotnet.microsoft.com/apps/aspnet

https://www.w3schools.com/sql/sql_injection.asp

https://www.w3schools.com/html/default.asp

Flowcharts and pseudocode

https://en.wikipedia.org/wiki/Flowchart

https://pseudocodeexamples.com/how-to-write-pseudocode/

https://www.code4example.com/pseudocode/pseudocode-examples/

Specification

Find the specification online: https://qualifications.pearson.com/en/qualifications/btec-nationals/computing-2016.html

The course that we are following is “Computing for Creative Industries” – this simply defines which units are sat. The content of units is identical regardless of which overall qualification you are working towards.