Programming Language Subset Summary of changes Summer 2025

Pearson Edexcel Level 1/Level 2 GCSE (9–1) in Computer Science Paper 2 – Application of Computational Thinking (1CP2/02)

Introduction

In this document, teachers and centres are informed about the changes made to the programming language subset (PLS) document from summer 2024 to summer 2025. The main changes detailed below are:

- Clarification of the two supported dimensions.
- Angle brackets amendments.
- Change to <string>.format in String subprograms table.
- Description update for built-in subprograms table.
- Spacing before bracket amendments.

The programming language subset (PLS) document version 6 for Summer 2025 can be found here.

Please see list of changes below.

Version 6 (Summer 2025)

Dimensions – Page 6

Further detail provided for the two supported dimensions shown in a table.

Dimensions

The number of dimensions supported by the PLS is two.

`<structure>[<first dimension>]`	`Accesses an item in a one-dimensional structure`
`<structure>[<first dimension>] [<second dimension>]`	`Accesses an item in a two-dimensional structure`

The PLS does not support ragged data structures. Therefore, in a list of records, each record will have the same number of fields.

Version 5 (Summer 2024)

Dimensions – Page 5

Dimensions

The number of dimensions supported by the PLS is two.

The PLS does not support ragged data structures. Therefore, in a list of records, each record will have the same number of fields.

<table>
<tr><th>Version 6 (Summer 2025)</th><th>Version 5 (Summer 2024)</th></tr>
<tr><td>Selection – Page 8
Angle brackets added for consistency.

<table><tr><td>if <expression>:
    <command></td><td>If <expression> is true, then <command> is executed.</td></tr></table></td><td>Selection – Page 7

<table><tr><td>if <expression>:
    <command></td><td>If <expression> is true, then command is executed.</td></tr></table></td></tr>
<tr><td>Iteration – Page 8
Angle brackets added for consistency.

<table><tr><td>for <id> in <structure>:
<command></td><td>Executes <command> for each element of <structure>, in one dimension.</td></tr></table></td><td>Iteration – Page 7

<table><tr><td>for <id> in <structure>:
<command></td><td>Executes <command> for each element of a data structure, in one dimension.</td></tr></table></td></tr>
<tr><td>Built-in subprograms table – Page 10
Angle brackets added for consistency.

<table><tr><td>input (<prompt>)</td><td>Displays <prompt> on the screen and returns the line typed in</td></tr></table></td><td>Built-in subprograms table – Page 9

<table><tr><td>input (<prompt>)</td><td>Displays the content of prompt to the screen and waits for the user to type in characters followed by a new line</td></tr></table></td></tr>
<tr><td>Built-in subprograms table – Page 10
Display replaced by screen in description.

<table><tr><td>print (<item>)</td><td>Displays <item> on the screen</td></tr></table></td><td>Built-in subprograms table – Page 9

<table><tr><td>print (<item>)</td><td>Prints <item> to the display</td></tr></table></td></tr>
</table>

String subprograms table - Page 12

Angle brackets added for consistency.

Changes to <string>.format

Subprogram	Description
`len (<string>)`	Returns the length of <string>
`<string>.find (<substring>, <start>, <end>)`	Returns the location of the first instance of <substring> in the original <string>, reading from left to right. <start> is the index to begin the find. The default is zero. <end> is the index to stop the find. The default is the end <string>. Returns -1, if not found.
`<string>.index (<substring>, <start>, <end>)`	Returns the location of the first instance of <substring> found in the original <string> as read from left to right. Raises an exception if not found. <substring> is required. <start> and <end> are optional. The default value for <start> is zero. The default value for <end> is the end of <string>.
`<string>.isalpha ()`	Returns True, if all characters are alphabetic A–Z
`<string>.isalnum ()`	Returns True, if all characters are alphabetic A–Z or digits 0–9
`<string>.isdigit ()`	Returns True, if all characters are digits 0–9, exponents are digits
`<string>.replace (<s1>, <s2>)`	Returns original <string> with all occurrences of <s1> replaced with <s2>
`<string>.split (<char>)`	Returns a list of all substrings in original <string>, using <char> as the separator
`<string>.strip (<char>)`	Returns original <string> with all occurrences of <char> removed from the front and back
`<string>.upper ()`	Returns original <string> in uppercase
`<string>.lower ()`	Returns original <string> in lowercase
`<string>.isupper ()`	Returns True, if all characters in <string> are uppercase
`<string>.islower ()`	Returns True, if all characters in <string> are lowercase
`<string>.format (<values>)`	Formats <values> and puts them into <string>. The content of <string> is described by symbols and placeholders.

String subprograms table – Page 11

Subprogram	Description
`len (<string>)`	Returns the length of <string>
`<string>.find (<substring>, <start>, <end>)`	Returns the location of the first instance of <substring> in the original <string>, reading from left to right. <start> is the index to begin the find. The default is zero. <end> is the index to stop the find. The default is the end of the string. Returns -1, if not found.
`<string>.index (<substring>, <start>, <end>)`	Returns the location of the first instance of <substring> found in the original <string> as read from left to right. Raises an exception if not found. <substring> is required. <start> and <end> are optional. The default value for <start> is zero. The default value for <end> is the end of the string.
`<string>.isalpha ()`	Returns True, if all characters are alphabetic A–Z
`<string>.isalnum ()`	Returns True, if all characters are alphabetic A–Z or digits 0–9
`<string>.isdigit ()`	Returns True, if all characters are digits 0–9, exponents are digits
`<string>.replace (<s1>, <s2>)`	Returns original string with all occurrences of <s1> replaced with <s2>
`<string>.split (<char>)`	Returns a list of all substrings in the original, using <char> as the separator
`<string>.strip (<char>)`	Returns original string with all occurrences of <char> removed from the front and back
`<string>.upper ()`	Returns the original string in uppercase
`<string>.lower ()`	Returns the original string in lowercase
`<string>.isupper ()`	Returns True, if all characters are uppercase
`<string>.islower ()`	Returns True, if all characters are lowercase
`<string>.format (<placeholders>)`	Formats values and puts them into the <placeholders>

<table>
<tr><th>Version 6 (Summer 2025)</th><th>Version 5 (Summer 2024)</th></tr>
<tr><td>Tips for using turtle - Page 15
Space before brackets added for consistency.

Add a line asking for keyboard input, such as input (), as the last line.</td><td>Tips for using turtle - Page 14

Add a line asking for keyboard input, such as input(), as the last line.</td></tr>
<tr><td>Turtle window and drawing canvas – Page 15
Space before brackets added for consistency.

<table><tr><td>turtle.Screen ()</td><td>Returns a variable to address the turtle window. Use with <window>.setup ().</td></tr></table></td><td>Turtle window and drawing canvas – Page 14

<table><tr><td>turtle.Screen ()</td><td>Returns a variable to address the turtle window. Use with <window>.setup().</td></tr></table></td></tr>
<tr><td>Turtle filling shapes – Page 16
Space before brackets added for consistency.

<table><tr><td><turtle>.end_fill ()</td><td>Call just after drawing the shape to be filled. You must call <turtle>.begin_fill () before drawing</td></tr></table></td><td>Turtle filling shapes – Page 15

<table><tr><td><turtle>.end_fill ()</td><td>Call just after drawing the shape to be filled. You must call <turtle>.begin_fill() before drawing.</td></tr></table></td></tr>
<tr><td>Turtle creation, visibility and movement – Page 16
Anticlockwise replaced by counterclockwise

<table><tr><td><turtle>.left (<degrees>)</td><td>Turns counterclockwise the number of <degrees></td></tr></table></td><td>Turtle creation, visibility and movement – Page 15

<table><tr><td><turtle>.left (<degrees>)</td><td>Turns anticlockwise the number of <degrees></td></tr></table></td></tr>
</table>

Turtle creation, visibility and movement – Page 16 Angle brackets added for consistency.	Turtle creation, visibility and movement – Page 15
<turtle>.showturtle () Makes <turtle> visible	<turtle>.showturtle () Makes the turtle visible