Name: Charles Brian Bett Unit: COMP 428 Reg No: S13/11573/15

1. Explain what an esoteric programming language (or esolang) is. [1]

Ref: (https://tomassetti.me/discovering-arcane-world-esoteric-programming-languages/

https://esolangs.org/wiki/Main_Page)

An esolang or esoteric programming language is a programming language that is designed to test the boundaries of a programming language or challenge the norms of language design.

2. There is some debate as to the usefulness of esoteric languages. Suggest two (2) reasons that the development of esoteric languages might be considered a waste of time (i.e. not useful), and two reasons that esoteric languages might be considered useful [4]

These are the reasons why they are useful:

- Most esoteric languages run on small complier which used less memory.
- They are used for educational purposes.

These are the reasons why they are not useful:

- Since they are considered a joke they are a waste of time to learn.
- Some a like Nothing are too hard to learn and take too much time to learn.

3. Choose any two esoteric programming languages. Describe the languages in terms of their designer(s), year of initial design, and their general syntactic and semantic characteristics. Provide a short example code snippet, to illustrate the language's general characteristics (you do not have to write the code yourself). [5]

- Whitespace

(Ref: https://web.archive.org/web/20151108084710/http://compsoc.dur.ac.uk/whitespace/tutorial.html)

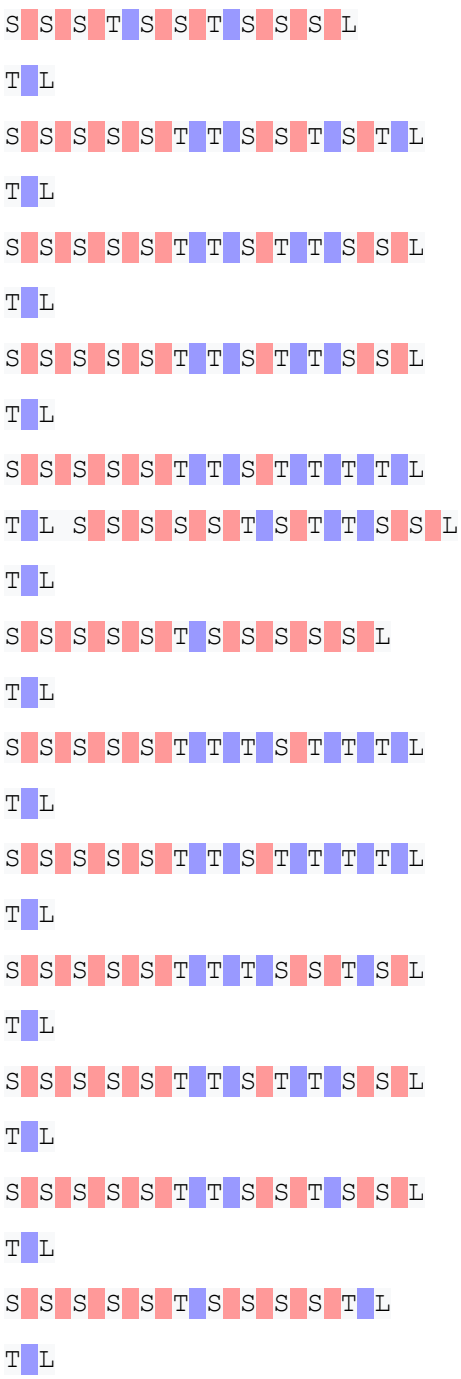
Initial design year: 2003

Commands are composed of sequences of spaces, tab stops and linefeeds. Code is written as an Instruction Modification Parameter (IMP) followed by the operation. The table below shows a list of all the IMPs in Whitespace:

IMP	Meaning
[Space]	Stack Manipulation
[Tab][Space]	Arithmetic

[Tab][Tab]	Heap Access
[LineFeed]	Flow Control
[Tab][LineFeed]	I/O

Sample Code: “Hello World!”



S S L L L

- Brainfuck

Ref: (https://esolangs.org/wiki/brainfuck)

Initial design year: 1993

The language consists of only eight simple commands and an instruction pointer. Brainfuck operates on an array of memory cells, also referred to as the tape, each initially set to zero. There is a pointer, initially pointing to the first memory cell. The commands are:

Command	Description
`>`	Move the pointer to the right
`<`	Move the pointer to the left
`+`	Increment the memory cell under the pointer
`-`	Decrement the memory cell under the pointer
`.`	Output the character signified by the cell at the pointer
`,`	Input a character and store it in the cell at the pointer
`[`	Jump past the matching `]` if the cell under the pointer is 0
`]`	Jump back to the matching `[` if the cell under the pointer is nonzero

Sample code: “Hello World!”

```
[
This program prints "Hello World!"
]
++++++++ Set Cell #0 to 8
[
      >++++ Add 4 to Cell #1; this will always set Cell #1 to 4
      [     as the cell will be cleared by the loop
```

```
      >++   Add 2 to Cell #2
      >+++ Add 3 to Cell #3
      >+++ Add 3 to Cell #4
      >+ Add 1 to Cell #5
      <<<<- Decrement the loop counter in Cell #1
]     Loop till Cell #1 is zero; number of iterations is 4
>+ Add 1 to Cell #2>+ Add 1 to Cell #3
>- Subtract 1 from Cell #4
>>+ Add 1 to Cell #6
[<] Move back to the first zero cell you find; this will be Cell #1
which was cleared by the previous loop
<- Decrement the loop Counter in Cell #0
]
>>. Cell #2 has value 72 which is 'H'
>---. Subtract 3 from Cell #3 to get 101 which is 'e'
+++++++..+++. Likewise for 'llo' from Cell #3
>>.   Cell #5 is 32 for the space
<-.   Subtract 1 from Cell #4 for 87 to give a 'W'
<.    Cell #3 was set to 'o' from the end of 'Hello'
+++.------.--------. Cell #3 for 'rl' and 'd'
>>+.  Add 1 to Cell #5 gives us an exclamation point
>++.  And finally a newline from Cell #6z
```

4. Consider the concept of multi-paradigm programming languages and answer the following questions:

(a) Identify one advantage associated with multi-paradigm programming languages. [1]

Multi-paradigm languages provide accurate and fast code writing, facilitate testing and debugging.

(b) Identify one disadvantage associated with multi-paradigm programming languages. [1]

They are complex thus hard to read.

(c) Identify a multi-paradigm programming language that combines the paradigms of functional and logic programming. [1]

Haskell

5. Consider the concept of a program that can manipulate itself as its data, and answer the following questions:

(a) Name the language feature that allows programs to manipulate themselves [1]

- metaprogramming

(b) Name one (1) language that provides native support for the above-mentioned feature. [1]

- Ruby

References

1. https://tomassetti.me/discovering-arcane-world-esoteric-programming-languages/
2. https://esolangs.org/wiki/Main_Page
3. https://en.wikipedia.org/wiki/Brainfuck
4. https://esolangs.org/wiki/brainfuck
5. https://web.archive.org/web/20151108084710/http://compsoc.dur.ac.uk/whitespace/tutorial.html
6. https://itnext.io/pros-and-cons-of-functional-programming-32cdf527e1c2
7. https://en.wikipedia.org/wiki/Haskell_(programming_language)
8. https://en.wikipedia.org/wiki/Metaprogramming