Compiler Construction

Explain a possible implementation method for Java-style or ML-style exceptions and handlers.

[8 marks]

Consider a simple arithmetic expression $e$ of abstract syntax:

$$e ::= x \mid n \mid e + e' \mid e - e' \mid e * e' \mid e/e' \mid -e$$

where $x$ ranges over a set of (global) variables, addressable by name, and $n$ ranges over integer constants. Write a procedure in pseudo-code or a language of your choice which takes an expression $e$ and prints (one-per-line) stack-machine instructions of the form

- `pushvar x`
- `pushnum n`
- `add` : pop two items and push their sum
- `sub` : pop two items and push their difference
- `mul` : pop two items and push their product
- `div` : pop two items and push their quotient
- `neg` : replace top item with its negation

which, when executed, have the net effect of pushing just the value of $e$ onto the stack. Each line of code emitted should contain a comment giving the number of items on the stack after its execution, thus the first `push` and the last instruction would both be commented with “1 item”.

[12 marks]