

HW1

1. A) Fortran was created by John Backus which first appeared in 1957. An important new feature that Fortran introduced is that Fortran was the first high-level programming language to exist.

Fortran Syntax:

$Y = A - \sin(B - C)$

B) LISP was designed by John McCarthy and developed by Steve Russell, Timothy P. Hart, and Mike Levin which first appeared in 1958. An important new feature was the introduction of recursion.

For example:

```
(defun factorial (n)
  (if (zerop n) 1
      (* n (factorial (1- n)))))
```

C) COBOL was designed by CODASYL and first appeared in 1959. An important new feature is that the syntax is more English-like.

For example:

MOVE x TO y.

D) Simula was designed by Ole-Johan Dahl and developed by Kristen Nygaard which first appeared in 1962. An important feature of this programming language is that this is considered the first object-oriented programming language.

For example:

```
Begin
  Class Hello;
    Virtual: Procedure print Is Procedure print;;
  Begin
  End;
  Hello Class Char (c);
    Character c;
  Begin
    Procedure print;
      OutChar(c);
    End;
  End
End
```

E) CLU was developed by Barbara Liskov and her students at MIT in 1973 which first appeared in 1975. An important feature of the language is that CLU introduced multiple assignments.

For example:

$x, y = y, x$
 $x, y, z = f(t)$

F) Smalltalk was designed and developed by Alan Kay, Dan Ingalls, Adele Goldberg, and others which first appeared in 1972. Smalltalk introduced classes where each class holds its private

state can communicate with itself and other objects through messages, and as it processes a message, it can send messages to both itself and other objects.

For example:

Object subclass: #Account.

2. Sec 1.8 1.1 (Java)

(a) A lexical error, detected by the scanner

Miss spelled keywords such as: 'if', 'while', 'for'.

(b) A syntax error, detected by the parser

A missing semicolon, ';'.
int a = 'hello';

(c) A static semantic error, detected by semantic analysis

int a = 'hello';

(d) A dynamic semantic error, detected by code generated by the compiler

int[] a = new int[5];

a[5] = 1;

(e) An error that the compiler can neither catch nor easily generate code to catch (this should be a violation of the language definition, not just a program bug)

int a;

if(false){

a = 5; //unreachable code

}

3. Sec 2.6 2.1

(a) $((\text{not}[\backslash"] | [\backslash \backslash | \backslash"]))^* "$

(b) $(\text{not}(\{ * * \} | * \{ \} *) | (* * | \{ \}))$

(c) octal integer: $0[0-7]^* [U|u|L|l|LL|ll|\epsilon]$

decimal integer: $[1-9]^* ([0-9]| [0-9].)[0-9]^* [U|u|L|l|LL|ll|\epsilon]$

hexadecimal integer: $0[x|X][0-9, a-f, A-F]^* [U|u|L|l|LL|ll|\epsilon]$

decimal floating-point: $[0-9]^* (([0-9]| [0-9].)^* | ([e|E][+|-|\epsilon] [0-9]^* [F|f|L|l|\epsilon]))$

hexadecimal floating-point:

$0[x|X] [0-9]^* (([0-9]| [0-9].)^* | \epsilon) [p|P][+|-|\epsilon] [0-9]^* [F|f|L|l|\epsilon]$

(d) $([0-9, a-f, A-f] ([_|\epsilon] [0-9, a-f, A-f])^*$

$[(\#([_|\epsilon] [0-9, a-f, A-f])^* . ([_|\epsilon] [0-9, a-f, A-f])^* \#) |\epsilon]$

$[e|E][+|-|\epsilon][0-9]^*)$

(e) $[0-9 | \#]^* [. (0-9 | \# | \epsilon)^* | \epsilon]$

(f) $\$ [*|\epsilon]^* (0 | [1-9] ([0-9] | [0-9] [0-9] | ([0-9][0-9][0-9])^*) |\epsilon) ([. [0-9][0-9] | \epsilon)$