Different Forms of Software

Steven W. Oxman

NCCA

IT Division

Software

- Logic that is run on computing machinery
  - Data Logic
  - Computational Logic
  - Networking Logic
  - Presentation Logic
- Otherwise Known As:
  - Computer Programs
  - Firmware
  - Computing Logic
  - Code

Types of Software

- Device Automation (sensors and actuators)
- Platforms (ship control systems)
- Communications (software controlled terminals)
- Business Software (HR Systems)
- Enterprise Resource Planning (Navy ERP)

Software Language Levels

- Machine Language
- Assembler Language
- High Level Language
- 4th Gen Language
- 5th Gen Language (aka, Artificial Intelligence)
- Specialty Levels (e.g., RICE)

Software Languages

- Machine Language
- Assembler Language (ULTRA)
- Fortran, COBOL, PASCAL, CMS-2, C
- Focus
- ADA, Prolog, LISP, C++, VB
- VBA, VBS, C#, JAVA, JAVAScript, PERL
- ABAP (SAP), Cold Fusion

Software Structures

- Straight Code
- Data Driven Code
- Object Oriented Code
- RICE Object Code

Other Software Classifiers

- Real Time
- Batch
- Interactive
- On-Demand

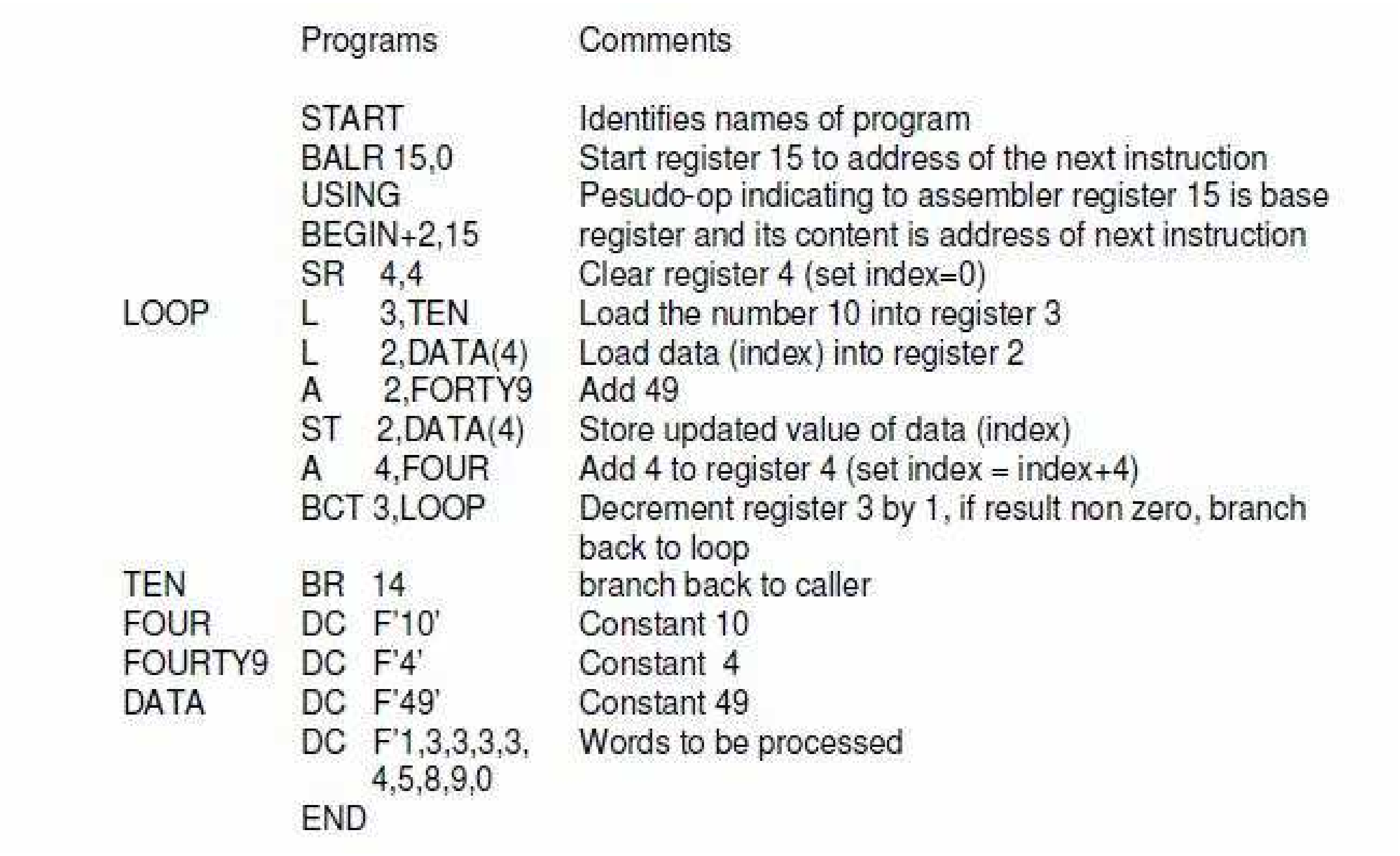
Other Software Classifiers in DON

- Tactical Software
- Non-Tactical Software
- Human Safety Certified Software
- Nuclear Safety Certified Software

Machine Code

100000010 01 00101	Load Memory 5 > R1
100000010 10 00101	Load Memory 5 > R2
1010000100 00 01 10	R1 + R2 > R0
100000100 00 00110	Store R0 > Memory 6

Assembler Code



```
              Programs          Comments

              START             Identifies names of program
              BALR 15,0         Start register 15 to address of the next instruction
              USING             Pesudo-op indicating to assembler register 15 is base
              BEGIN+2,15        register and its content is address of next instruction
              SR    4,4         Clear register 4 (set index=0)
LOOP          L     3,TEN       Load the number 10 into register 3
              L     2,DATA(4)   Load data (index) into register 2
              A     2,FORTY9    Add 49
              ST    2,DATA(4)   Store updated value of data (index)
              A     4,FOUR      Add 4 to register 4 (set index = index+4)
              BCT 3,LOOP        Decrement register 3 by 1, if result non zero, branch
                                back to loop
TEN           BR   14           branch back to caller
FOUR          DC   F'10'        Constant 10
FOURTY9       DC   F'4'         Constant  4
DATA          DC   F'49'        Constant 49
              DC   F'1,3,3,3,3, Words to be processed
                   4,5,8,9,0
              END
```

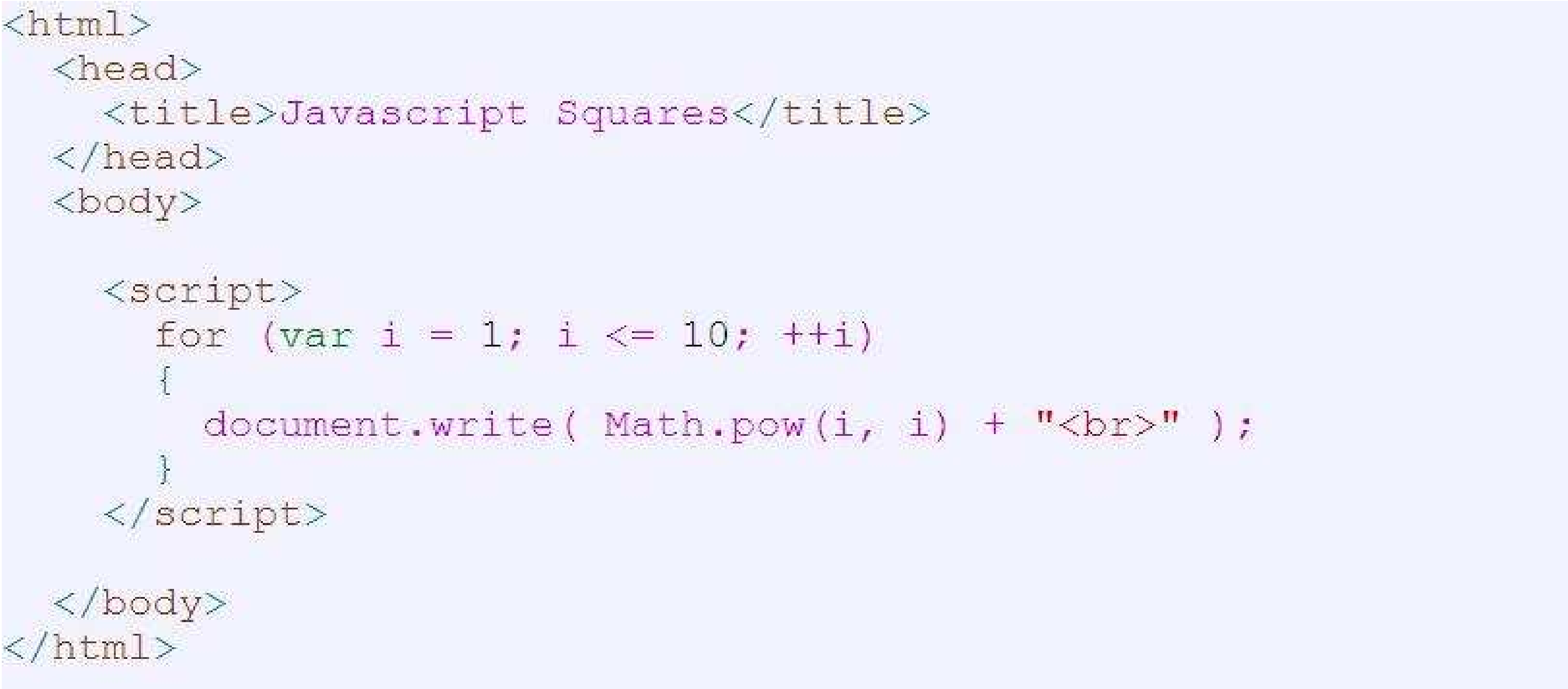
Fortran Code

```
   PROGRAM SQUARE
   DO 15, I = 1,10
   WRITE(*, *) I*I
15 CONTINUE
    END
```

C# Code

```
using System;
class Squares1 {
    static void Main() {
        for (int i=1; i<=10; i++) {
            Console.WriteLine("{0} ", i*i);
       }
   }
}
```

JavaScript Code



```
<html>
  <head>
    <title>Javascript Squares</title>
  </head>
  <body>

    <script>
      for (var i = 1; i <= 10; ++i)
      {
        document.write( Math.pow(i, i) + "<br>" );
      }
    </script>

  </body>
</html>
```

LISP Code

```
(dotimes (i 10)
  (format t "~D " (expt i 2)))
```

PERL Code

```
print join(" ", map { $_ ** 2 } 1..10), "\n";
```

ABAP (SAP) Code

```
start-of-selection.
classname_select = p_class.
classtype_select = 'OT'.
objkey_select = p_objct.
PERFORM f_bds_call_navigator.

at line-selection.
PERFORM f_line_selection.


FORM f_bds_call_navigator.
* Function : BDS_CALL_NAVIGATOR
* -- data declaration -- *
DATA: i_connections LIKE bdn_con OCCURS 0 WITH HEADER LINE,
calling_signa LIKE bapisignat OCCURS 0 WITH HEADER LINE,
i_count LIKE sy-index,
i_gui_type LIKE bapibds01-type,
web_excluding LIKE bdn_fkt OCCURS 1 WITH HEADER LINE.
```

ABAP (SAP) Subroutine Code

```
REPORT demo_mod_tech_example_2.
DATA: num  TYPE i VALUE 5,
      fac  TYPE i VALUE 0.
PERFORM fact USING num CHANGING fac.
WRITE: / 'Factorial of', num, 'is', fac.
FORM fact
       USING value(f_num) TYPE i
       CHANGING f_fact    TYPE i.
 f_fact = 1.
    WHILE f_num GE 1.
     f_fact = f_fact * f_num.
     f_num = f_num - 1.
    ENDWHILE.
ENDFORM.
```

Software Issues for Costers

- Software Use (real time versus office use)
- Software Testing (path testing vs sim/stim)
- Software Language (generation?)
- Special Requirements (e.g., nuc cert)
- Code Counting (whose counting rules)
- Straight code versus object oriented code
- In-code documentation
- DBMS vs Data Tables
- …