

Intermediate RPG IV Program Standards

- Code the beginning of the program as follows:

```
//-----
// Program      : R9999XXX (9999 is assignment no, XXX is your initials)
// Name         : Your Name
// Date         : 10/05/2009 (Today's date)
// Description   : This program reads a file.
//-----
```

- Comment lines to separate each spec

```
//-----F Spec-----
```

- Document before mainline

```
//-----
// Main Processing Routine
// Description of mainline
//-----
```

- Document to explain code that isn't apparent by reading the source.

```
taxDue = purchase * SALES_TAX // tax due is purchase times 0.05
```

- Document before each subroutine

```
//-----
// Description of subroutine as a whole
//-----
```

- Document before each procedure

```
//-----
// Description of return values
// Description of parameters
//-----
```

- Naming Conventions:

- Constants all CAPS(SALES_TAX)
- RPG-reserved words all CAPS(READ) or Start with Uppercase(Read)
- Variables should be CamelCase(CustNo) or Hungarian notation(custNo)
- Variables names can be from 10 to 14 characters long
- Subroutine names should be Verb-Object(PrintDetail)
- Check off "R99C9XXX"
- Program "R99P9XXX"

- Correct standard heading on reports

```

      1          2          3          4          5          6
      12345678901234567890123456789012345678901234567890..
Line#1 R9999XXX                               TITLE          PAGE   XXX0
Line#2 XX/XX/XXXX                               (last name) USERNAME

Line#4 Column Heading1      Column Heading 2      Etc

Line#6   Detail Line
Line#7   Detail Line

```


Intermediate RPG IV Program Standards

```
//-----  
// Program      : R9999XXX (9999 is assignment no, XXX is your initial  
// Name         : Your Name  
// Date         : 10/05/2009 (Today's date)  
// Description  : This program produced a year-to-date sales report.  
//-----  
  
// -----F Spec -----  
  
// -----I Spec -----  
  
// -----D Spec -----  
  
/Free  
//-----  
// Main Processing Routine  
//   Print headings, priming read, check for EOF, check for overflow,  
//   print detail and read next record.  
//-----  
  
    Except Headings;                // Print headings  
    Read Customers;                // Priming read  
  
    Dow Not %Eof(Customers);        // Check for EOF  
  
    If *Inof;                       // Check for overflow  
        Except Headings;  
        *Inof = *Off;  
    ENDIF;  
  
    Except Detail;                 // Print detail  
    Read Customers;                // Read next record  
    ENDDO;  
  
    *Inlr = *On;  
    Return;  
  
//-----  
// Subroutine  
//   This subroutine will do something.  
//  
//-----  
  
/End-Free  
  
// -----O Spec-----
```