**CSCU 360 USING, DROP and Dummy Sections**

**USING statement**

Format: USING label,R

R contains an address

The USING statement tells the assembler to associate the address in R with label

It also tells the assembler which register to use when converting implicit addresses to explicit addresses.

A USING statements "reach" extends 4096 bytes.

**DROP statement**

Format: DROP R1,R2,...,Rn

Ends the domain of a USING statement

The DROP informs the assembler that registers R1,R2,...,Rn are no longer to be associated with label

or

that the specified register is not supposed to be used to convert implicit addresses to explicit addresses.

A USING statement is in effect until a DROP statement is encountered. Any routine coded below a DROP statement will not know which register to use as a base register unless it contains a USING statement of its own.

**Dummy SECTions**

A dummy section is used to specify a format that can be associated with a particular area in storage without producing any object code

A dummy section begins with:

Format: label DSECT

The end of a dummy section is signaled by the occurrence of a CSECT statement, another DSECT statement, or an END statement.

An example DSECT:

TABELEM DSECT

STCKNUM DS F

ARTIST DS CL24

TITLE DS CL24

INSTOCK DS F

PRICE DS F

The above DSECT specifies the format for a table element. The labels STCKNUM, ARTIST, etc can be used rather than displacements

Before a DSECT can be used, a USING statement must be coded.

USING TABELEM,R3

**Before DSECTs**

LM R3,R5,0(R1)

\*

\* R3 -> address of table

\* R4 -> address of NAV storage area

\* R5 -> address of input buffer

\*

XREAD 0(,R5),80

DO1 BL ENDDO1

XDECI R6,0(,R5)

ST R6,0(,R3)

MVC 4(24,R3),7(R5)

MVC 28(24,R3),32(R5)

XDECI R6,57(,R5)

ST R6,52(,R3)

XDECI R6,61(,R5)

ST R6,56(,R3)

LA R3,60(,R3)

XREAD 0(,R5),80

B DO1

ENDDO1 DS 0H

ST R3,0(,R4)

**After DSECTs**

USING TABELEM,R3

LM R3,R5,0(R1)

\*

\* R3 -> address of table

\* R4 -> address of NAV storage area

\* R5 -> address of input buffer

\*

XREAD 0(,R5),80

DO1 BL ENDDO1

XDECI R6,0(,R5)

ST R6,STCKNUM

MVC ARTIST(24),7(R5)

MVC TITLE(24),32(R5)

XDECI R6,57(,R5)

ST R6,INSTOCK

XDECI R6,61(,R5)

ST R6,PRICE

LA R3,60(,R3)

XREAD 0(,R5),80

B DO1

ENDDO1 DS 0H

ST R3,0(,R4)

DROP R3

A second DSECT similar to the one below can be added to make the program even more readable:

INPUT DSECT

INSTKNM DS CL6

DS C

INART DS CL24

DS C

INTITLE DS CL24

DS C

INAMT DS CL3

DS C

INPRICE DS CL4

DS CL15

**With the second DSECT**

USING TABELEM,R3

USING INPUT,R5

LM R3,R5,0(R1)

\*

\* R3 -> address of table

\* R4 -> address of EOT storage area

\* R5 -> address of input buffer

\*

XREAD INPUT,80

DO1 BL ENDDO1

XDECI R6,INSTKNM

ST R6,STCKNUM

MVC ARTIST(24),INART

MVC TITLE(24),INTITLE

XDECI R6,INAMT

ST R6,INSTOCK

XDECI R6,INPRICE

ST R6,PRICE

LA R3,60(,R3)

XREAD INPUT,80

B DO1

ENDDO1 DS 0H

ST R3,0(,R4)

DROP R3,R5