Using External Subroutines

One of the productive elements of computer software design is the use of subroutine(s) to reduce the amount of coding a programmer has to do by reducing repetitive instruction sequences. Once a subroutine(s) has been designed, it can then be made available for further use by other programs. These wind up forming a library of subroutines which one or more programmers can use within an application. This is accomplished by collecting all the common subroutines in one program and creating a single "object module" and having it linked together with your program whereby the two get merged together and it effectively appears as one.

The individual subroutines within the subroutine object module can be accessed by your program by using a assembler directive for each piece of code you use that's borrowed from an object module, as follows;

```
EXTRN     <subroutine_label>:FAR
```

Similarly, you also need to make the following addition to the ASSUME directive;

```
ASSUME    cs:code, ds:data, ss:stak
```

You also have to make sure that your program has an area of memory called a stack segment. Its inserted in your program after your data segment and before the END directive in the following manner;

```
stak      SEGMENT   STACK
          DW        256 dup (0)
stak      ENDS
```

The first sample program sampl1.asm shows how this is achieved. Also, to make use of these subroutines, you have to, as stated above, link your program with the appropriate object module like this;

```
DOS> ML  /c  /Fl  <your_pgm>
DOS> LINK  <your_pgm.obj>+pcio.obj, , , , ,
```

Below are the subroutines available to you to accomplish simple input and output on the PC and contained in the PCIO.OBJ object module.

Subroutine	Description	Entry	Exit
getbyte	Inputs 2 valid hex numbers from the keyboard. None hex no.'s are ignored.	None	**AL** - contains hex byte value
getword	Inputs 4 valid hex numbers from the keyboard. Non-hex no.'s are ignored.	None	**AX** - contains the hex word value.
outbyte	The contents of AL are printed on the screen as printable ASCII alphanumeric characters.	**AL** - contains the hex byte value to be sent to screen as ASCII	None
outword	The contents of AX are printed on the screen as printable ASCII characters.	**AX** - contains the hex word value to be sent to screen as ASCII	None
getc	Inputs a ASCII character from the keyboard.	None	**AL** - contains equivalent hex value of the ASCII character
outc	The ASCII equivalent character of the hex value contained in AL is sent to the screen.	**AL** - contains the hex value (ASCII value) to be sent to screen	
outstr	The ASCII byte values, which make up some character message array are sent to screen. The array must end with a byte value of 04h.	**DI** - contains the address of the ASCII character string.	None
newline	Sends a line feed and carriage return to the screen.	None	None