Program Format

A complete assembly source code contains the following sections:

- Data section declares initialized data
- BSS (Block Started by Symbol) section declares uninitialized data
- Text section contain code

Comments

- Comments start with semicolon (**;**)
- Comments can start at the beginning of line or after an instruction

Numeric Values

- Hex number must start with 0x for example, 0x7F, 0x12, 0xAB
- Default base is decimal, no special is required

Constants

- Constants can be defined with **equ** directive for example,

```
SIZE    equ     10000
```

- Constants are substituted for their defined values during assembly process.

Program Format

Data Section

- Data section begins with "`section .data`" directive
- Variable names must begin with a letter and followed by letters or numbers
- Variable declaration format

  <variableName> <dataType> <initialValue>

- Data types are as follows:

Declaration	
`db`	8-bit variable(s)
`dw`	16-bit variable(s)
`dd`	32-bit variable(s)
`dq`	64-bit variable(s)
`ddq`	128-bit variable(s) → integer
`dt`	128-bit variable(s) → float

Program Format

Examples:

```
bVar    db    10                ; byte variable
cVar    db    "H"               ; single character
str     db    "Hello World"     ; string
wVar    dw    5000              ; word variable
dVar    dd    50000             ; 32-bit variable
arr     dd    100, 200, 300     ; 3 element array
flt1    dd    3.14159           ; 32-bit float
qVar    dq    1000000000        ; 64-bit variable
```

The value specified must fit the specify data type. For example:
a byte variable can contain value 0-255

Program Format

BSS Section

- BSS section begins with "`section .bss`" directive
- Variable names must begin with a letter and followed by letters or numbers
- Variable declaration format

  <variableName> <reserveType> <repetition>

- Data types are as follows:

Declaration	Meaning	Size
resb	Reserve byte	8-bit variable
resw	Reserve word	16-bit variable
resd	Reserve double word	32-bit variable
resq	Reserve quad word	64-bit variable
resdq	Reserve double-quad word	128-bit variable

Program Format

Examples:

```
bArr        resb        10          ; 10 element byte array
wArr        resw        50          ; 50 element word array
dArr        resd        100         ; 100 element double array
qArr        resq        200         ; 200 element quad array
```

The variables allocated above are not initialized to any specific value.
There are always a value at allocated memory space.

Program Format

Text Section

- Text section begins with "**`section .text`**" directive
- This section contains entry point of program:

```
global _start
_start:
```

- Each line in this section contains:

[label] <Instruction> <[operands]>

- Label(s) are optional and used as reference location in text section
- Labels must start with letter and followed by letters or numbers and closed with colon (**:**)

Example

```
;********* Simple example program *********
section   .data
;--------- Define constants --------------
EXIT_SUCCESS  equ    0   ;successful operation
SYS_exit      equ    60  ;call code for terminate

;--------- Define variables --------------
BVar          db     17
Wvar          dw     9000
Dvar          dd     50000

section   .text
global    _start
_start:
          mov    al, byte [Bvar]
          mov    bx, word [Wvar]
          mov    ecx, dword [Dvar]
exit:
          mov    rax, SYS_exit
          mov    rdi, EXIT_SUCCESS
          syscall
```