Awesome! Here’s the **full, neatly formatted content** ready to copy-paste into your favorite word processor or markdown editor. You can then save/export it as a PDF.

**Microprocessor, Microcontroller, and Assembly Language – Class Notes**

**1. Basic Concepts**

**Comment Syntax**

* Use ; to write comments in assembly language.
* Comments explain each line and do not affect program execution.

**2. Standard Assembly Program Structure (8086)**

.MODEL SMALL ; Defines program memory model (can be TINY, SMALL, MEDIUM, LARGE)

.STACK 100H ; Allocates 256 bytes (100H) for the stack

.DATA ; Section to declare and define variables (constants, strings, etc.)

.CODE ; Section for writing actual instructions (code)

**3. Common Instructions and Their Purpose**

**Printing a Character**

MOV AH, 2 ; Function to display a character

MOV DL, 'A' ; Character to display is stored in DL

INT 21h ; Calls DOS interrupt to print the character

Explanation:

* AH = 2 sets the function code for printing a single character.
* DL holds the character to be printed.
* INT 21h triggers the interrupt to execute the function.

**Printing a String**

MOV AH, 9 ; Function to display a string

LEA DX, msg ; Load address of string (msg) into DX

INT 21h ; Execute interrupt to print the string

Important:

* Strings must end with $ for this method to work.
* Strings are defined in the .DATA section using db:

msg db "Hello World$"

**Taking Input from User**

MOV AH, 1 ; Function to read a single character from input

INT 21h ; Waits for user input

The input character is stored in AL register.

**Displaying the Character Just Inputted**

MOV AH, 1 ; Get user input

INT 21h

MOV DL, AL ; Move input character to DL

MOV AH, 2 ; Function to display a character

INT 21h

**4. Exiting the Program**

MOV AH, 4CH ; Function to terminate program and return to DOS

INT 21h

**5. Extra Notes**

* Printing is always done via register DL, either directly (for a character) or indirectly (via DX for strings).
* Binary to Hexadecimal conversion method was mentioned, details to be covered later.

**6. Sample Assembly Program: Print a String**

.MODEL SMALL

.STACK 100H

.DATA

msg db "Hello World$" ; Define string to print

.CODE

MAIN PROC

MOV AX, @DATA ; Load address of data segment

MOV DS, AX ; Initialize DS with data segment

MOV AH, 9 ; Function to print string

LEA DX, msg ; Load address of msg into DX

INT 21h ; Print the string

MOV AH, 4CH ; Exit the program

INT 21h

MAIN ENDP

END MAIN

**7. Summary of DOS Interrupt 21h Functions**

| **Function** | **AH Value** | **Purpose** | **Notes** |
| --- | --- | --- | --- |
| Print Char | 02H | Print single character in DL | DL must have char to print |
| Input Char | 01H | Accept one character input | Result stored in AL |
| Print String | 09H | Print string ending with $ | Address must be in DX |
| Exit Program | 4CH | Exit to DOS | Program ends execution |

**8. What Are AH, AX, DL, DH, etc.?**

**Registers Explained**

Registers are **small, fast storage inside the CPU** used for temporary data during execution.

**The 8086 General Purpose Registers**

| **Full Register** | **16-bit** | **High Byte (8-bit)** | **Low Byte (8-bit)** | **Use** |
| --- | --- | --- | --- | --- |
| AX | Yes | AH | AL | Accumulator (math, I/O) |
| BX | Yes | BH | BL | Base (addressing) |
| CX | Yes | CH | CL | Count (loops, shifts) |
| DX | Yes | DH | DL | Data (I/O operations) |

**Specific Registers**

| **Register** | **Size** | **Description** |
| --- | --- | --- |
| AX | 16-bit | Accumulator used for arithmetic and data operations |
| AH | 8-bit | High byte of AX; stores function code for interrupts |
| AL | 8-bit | Low byte of AX; stores input/output data |
| DX | 16-bit | Data register for I/O and multiplication/division |
| DH | 8-bit | High byte of DX; sometimes used in character/address |
| DL | 8-bit | Low byte of DX; often stores character to output |

**Example Usage**

MOV AH, 2 ; Function to print a character

MOV DL, 'H' ; Character 'H' to print

INT 21h ; Call DOS interrupt

* AH = 2 → Function code for print character
* DL = 'H' → Character to print
* INT 21h → Call interrupt to perform the action

**Summary Table**

| **Register** | **Meaning** | **Common Use** |
| --- | --- | --- |
| AX | Accumulator | Math and data operations |
| AH | High byte of AX | DOS function codes |
| AL | Low byte of AX | Input/output data |
| DX | Data register | I/O and addressing |
| DL | Low byte of DX | Character output |
| DH | High byte of DX | Occasionally used |

**End of Notes**

If you want, I can also help with:

* Formatting tips for exporting PDF
* Creating diagrams for registers
* Sample programs for input/output in assembly

Just ask!