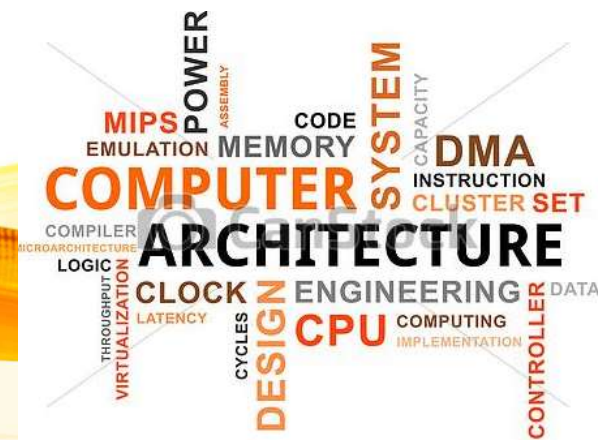


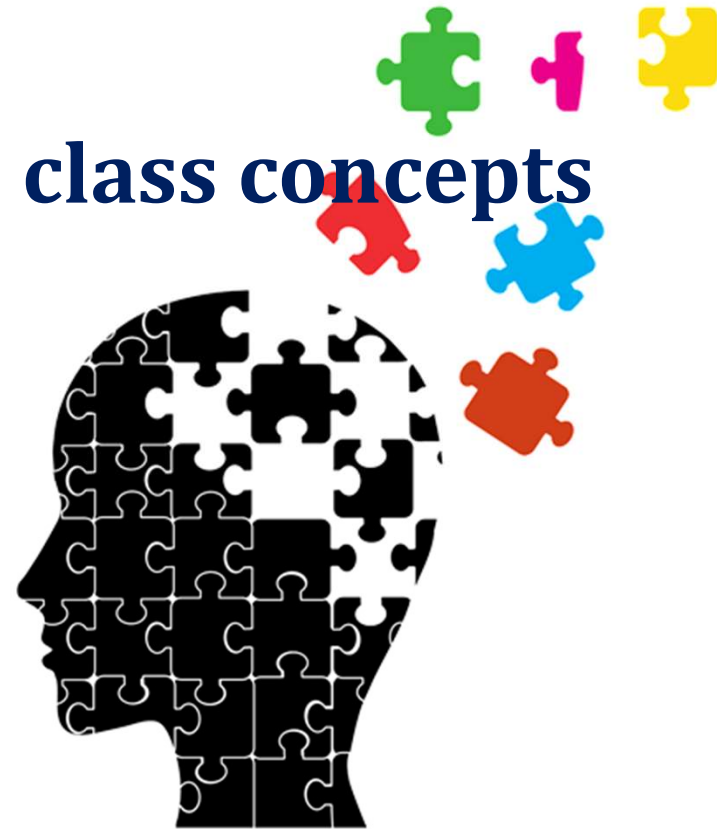
UNIT I

BASIC STRUCTURE OF COMPUTERS

Functional units – Basic operational concepts – Bus Structures – Performance – Memory locations and addresses – Memory operations – Instruction and Instruction sequencing – Addressing modes – **Assembly language** – Case study : RISC and CISC Architecture.



Recall the previous class concepts



Assembly language

- Assembly language is a type of programming language that communicates with the hardware of a computer.
- Hardware from different manufacturers uses machine language, like binary or hexadecimal characters, to perform tasks.

Assembly language

- Assignment Statement is $f = (g + h) - (i + j)$. What is the compiled MIPS code?

f,g,h,i,j is assigned to r0,r1,r2,r3,r4

Temp register r5,r6

```
add r5, r1, r2
add r6, r3, r4
sub r0, r5, r6
```

Assembly language

Convert the following C Language into MIPS Assembly
Language $A[30] = h + A[30]$

```
lw $t0, 32($s4) # load word
    add $t0, $s2, $t0
sw $t0, 32($s4) # store word
```

Index 8 requires offset of 32

Assignment statement is

```
g = h + A[8];
```

```
A[12] = h + A[8];
```

```
lw $t0, 8($s3) # load word  
add $s1, $s2, $t0
```

```
lw $t0, 32($s3) # load word  
add $t0, $s2, $t0  
sw $t0, 48($s3) # store word
```

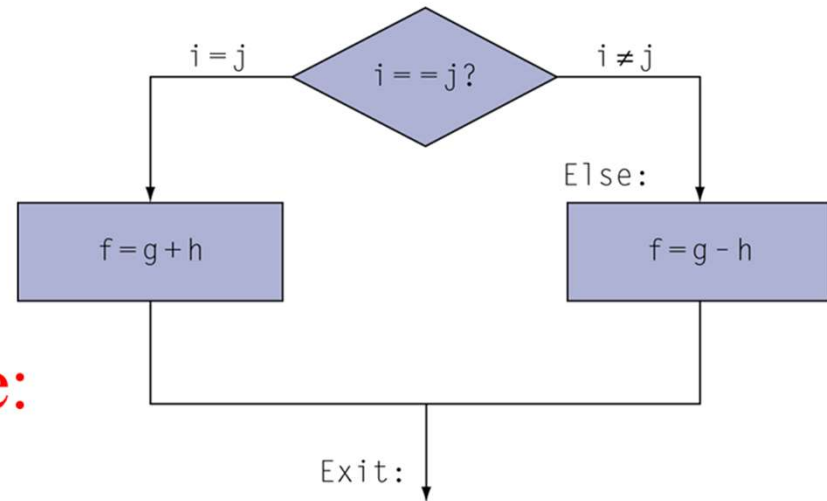


C code:

```
if (i==j)
    f = g+h;
else f = g-h;
- f, g, ... in $s0, $s1, ...
```

Compiled MIPS code:

```
beq $s3, $s4, Else
add $s0, $s1, $s2
j Exit
Else: sub $s0, $s1, $s2
Exit: ...
```





sns
INSTITUTIONS

