Union assignments

Mandatory

1. Refer the code below and comment on size of the given structure considering

a. Structure as union

b. Structure as struct

c. arr

d. uarr

A screenshot of a computer code

Description automatically generated

**Union Job** uses less memory (32 bytes for each element) because all members share the same space.

**Struct Job** uses more memory (56 bytes for each element) due to the need for separate memory for each member and potential padding for alignment.

2. Refer Job datastructure in Q#1 above. Using uarr, perform below operations.

a. Read and store salary

b. Read and store workerNo

Comment on values of output if salary and workerNo are printed in order. Justify your statement.

A computer screen shot of a black screen

Description automatically generated

A screen shot of a computer

Description automatically generated

**Union Behavior:** In a union, all members share the same memory location, and only one member's data can be stored at a time. Writing a value to one member will overwrite the data stored by other members.

**Memory Overlap:** When we assign uarr[0].salary = 50000.0 and later assign uarr[0].workerNo = 12345, we are effectively overwriting the memory previously used to store the salary with the value of workerNo. This is why the printed salary is not the expected 50000.0 but instead the value 12345.000000, which is just the int 12345 interpreted as a float.

3. Refer Job datastructure in Q#1 above. Assume that myvar is a structure variable. If I need to place 2 bytes (i.e 0x0102) as ucount using a char \*ptr then list all possible statements that can be used in \_\_\_\_\_.

[Let solutions include cases such as

i. using base address of ucount

ii. using relative address of ucount w.r.t to base address of myvar]

A white background with purple text

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

