Common error

1. In case of nested for loop change the int i , j

For(int I=0;I<1;I++)

For(int j=0;j<2;j++)

1. In case of object define from struct

student students[100] -> struct name + name of object

Uses of struct name is 1. give struct name 2. define object like upper ex

1. += (true) + = (false)
2. Dont forget function name -> void name ()
3. Object instiallization -> object name = { } ; like array
4. NEVER NEVER equate 2 char array ONLY ONLY use strcpy
5. Array passed to function using name -> fun(array name (pointer to the first element)) fun({1,2,3,4}) (false)
6. NEVER NEVER forget the assesing way to struct element struct name . element
7. don’t forget name of pointer -> address

\*name of pointer -> coqntent

& حاجه -> address of the حاجه

1. Any function use specific struct and use it must be declared after the struct
2. Cin char of array using
3. gets(array\_name);
4. Cin.get(array\_name , number of taken char) -> to avoid over flow
5. Cin string
6. Getline(cin,string\_name);

Quiz \_ 1 rev

1. Ascii a -> 97 z -> 122
2. Ascii ‘a’- ‘A’ = 32
3. don’t print any value before instialling it because it will hold garbage

Int arr[100] ; arr[0] = 0; cout<<arr[1] ; -> garbage value

1. When print uninstiallized varible it gives garbage value not compilar error
2. (\*) value of the address pointed to
3. \*pt = x false bec pt don’t pointe on any address
4. Pt = &x true
5. \*pointer cant hold address of another pointer

\*\*pointer must be used instead

1. Pointer store address its size determined by the machine

4\_byte -> 32 bit machine

8\_byte -> 64 bit machine

1. Char arr[] =”word”; size = 5

Char arr[] = {‘w’ ‘o’ ‘r’ ‘d’} size = 4

1. When cout char array it is printed till read \0 null operator
2. Char array pointer is the only pointer that can take value and this value is stored in the ROM so cant be modified

Ex) Char \* ptr =” eece”;

Cout<< ptr << \*ptr; -> eece e

| **Expression** | **Meaning** | **What it does** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| \*pt + 1 | **Dereference first, then add** | Takes value at pt, adds 1 to it |

|  |  |  |
| --- | --- | --- |
| \*(pt + 1) | **Pointer moves first, then dereference** | Moves to the next element, then gets value |

1. cout the \*(nullpointer) -> giver error if not nullpointer it will gives garbage value