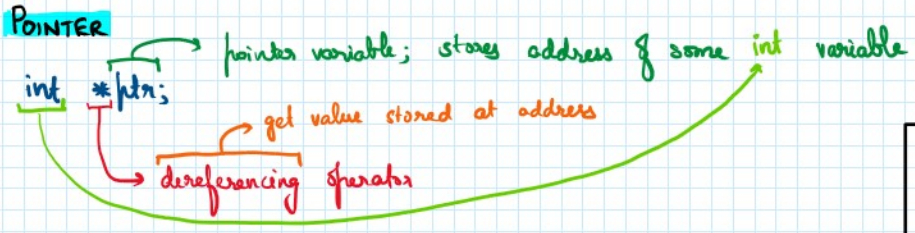
4. Pointers, Comma Operator

21 February 2024 09:32

POINTER



> **NOTE:**
> Format specifier for address : %p / %x

> **NOTE:**
> Size of pointer remains the same regardless of its datatype.
> It is implementation dependent.

```
int a = 2;
ptr = &a;
printf("%p", ptr);   → prints address
printf("%d", *ptr);  → prints value (2)
*ptr = 5;            → "a" value is set to 5
*ptr = &a;           → raises a warning in compilation. If program is run, address is stored in a as an integer (or whatever type was defined)
```

Why different pointers based on datatype?

When using *ptr to fetch the value at an address, the number of bytes fetched depends on the datatype allocated. For example:

```
char *ptr = a;   → fetches only one byte
int  *ptr = a;   → fetches four bytes
```

COMMA OPERATOR

- Sequence point operator : First LHS is evaluated; result and any sideeffects are disregarded when control goes to RHS.

```c
int a = 1, b = 2, c = 3, result;

result = (a, b, c);

printf("Result: %d\n", result);

return 0;
```

```
Result: 3
```

```c
#include <stdio.h>

int main()
{

    int a = 5;
    int b = 0;
    int c;
    c = (++a, b++, a++, b--, a++); // comma operator
demonstration with increment and decrement
    printf("a = %d, b = %d, c = %d\n", a, b, c);
    return 0;
}
```

```
a = 8, b = 0, c = 7
```