بسم الله الرحمن الرحيم

التاريخ: / / الموضوع

* Typedef : Keyword used to make your own datatype.

by providing an existed data type a new name.

ex.1 : typedef int Integer;

- It can be used with primitive data types, structs and enums

ex.2 :
```
typedef struct Student {
    char *name;
    int id;
} S;
int main(void) {
    S s1;
    s1.name = "Ahmed";
}
```

ex.3 :
```
typedef int* *ptr;

int main(void) {
  int n = 30;
  ptr p = &n;
  printf("%d", *p);
}
```
Output : 30

* Unions: User-defined data type contains different type of data types, Unlike structs: they share the same memory location

ex.1:
```
typedef union {
    int Integer;
    float Float;
    char Char;
} Joker;
```

Joker.Integer ← to store int value
Joker.Float ← " " float "
Joker.Char ← " " char "
The great part that they share the same memory location

And that means memory saving and effieiency

☆ Only one member can contain data at the same ~~by~~ time

---

* Enums:

ex:
```
typedef enum direction {
    ~~East~~ EAST,  → by default 0
    NORTH,  → " " 1
    WEST,  → " " 2
    SOUTH  → " " 3
} Direction
```

* Bit fields: use memory efficiently when we know that
  - the value of a field will never exceed a limit
  - → used when the storage of program is limited

- Data type of bit field must be integer

ex. 1:
```
typedef struct date {
    unsigned int d: 5;
    unsigned int m: 4;
    int y;
} Date;

int main() {
    Date dt = {31, 12, 2014};
    printf("%d/%d/%d", dt.d, dt.m, dt.y);
}
```
Output: 31/12/2014 ✓

* Out of Range value:

compilation error: Data trunction [-woverflow] (#)