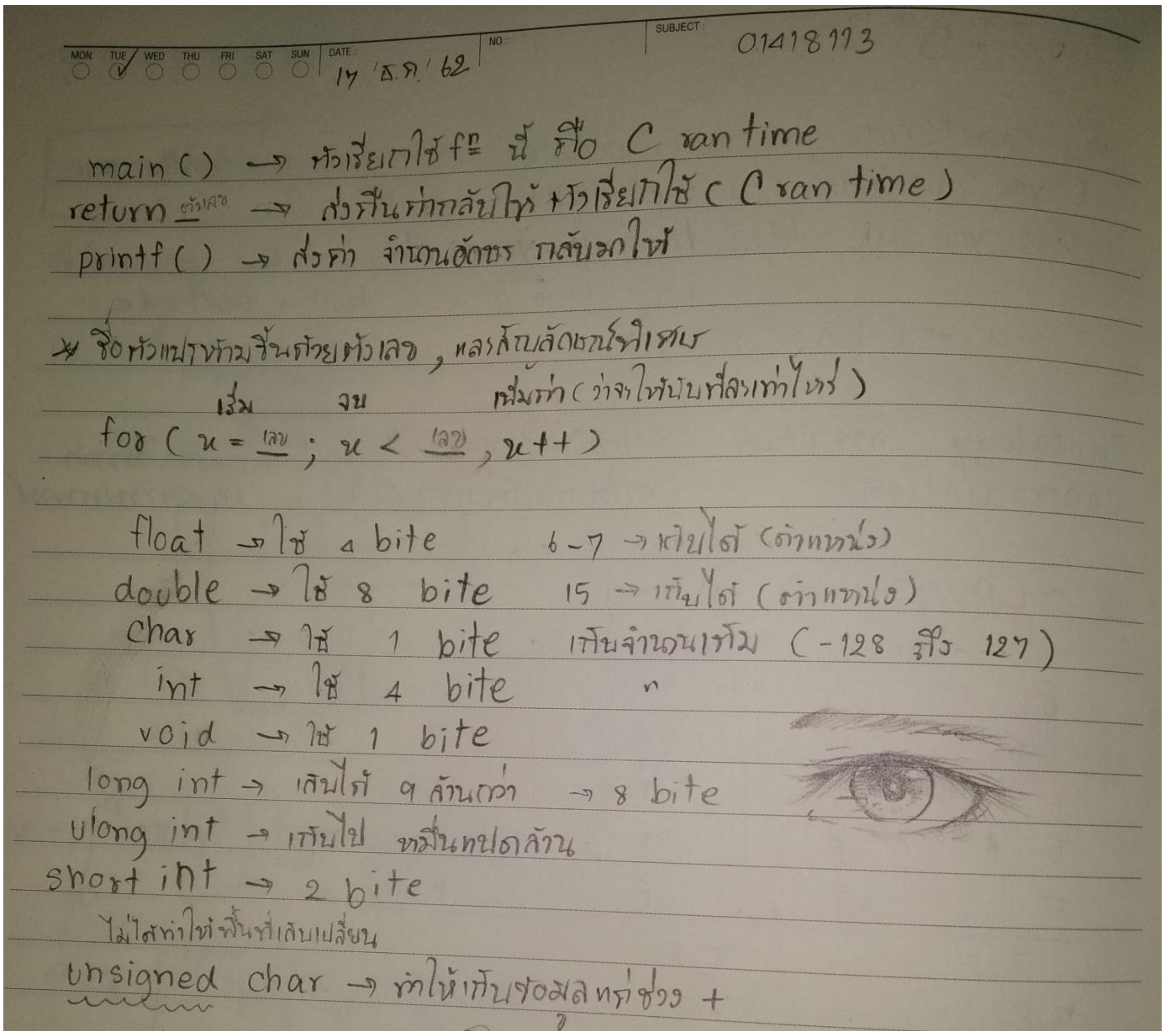
Week 2 : Type, Operators and Expression



MON [✓ TUE] WED THU FRI SAT SUN DATE : 17 'ธ.ค.' 62 NO : SUBJECT : 01418113

main ( ) → ตัวเรียกใช้ fⁿ นี้ คือ C ran time

return ตัวเลข → ส่งคืนค่ากลับให้ ตัวเรียกใช้ (C ran time)

printf ( ) → ส่งค่า จำนวนอักษร กลับมาให้

* ชื่อตัวแปรห้ามขึ้นด้วยตัวเลข , หรือสัญลักษณ์พิเศษ

```
        เริ่ม        จบ            เพิ่มค่า ( ว่าจะให้นับทีละเท่าไหร่ )
for ( x = เลข ; x < เลข , x++ )
```

float → ใช้ 4 bite 6-7 → เก็บได้ (ตำแหน่ง)

double → ใช้ 8 bite 15 → เก็บได้ (ตำแหน่ง)

char → ใช้ 1 bite เก็บจำนวนเต็ม (-128 ถึง 127)

int → ใช้ 4 bite "

void → ใช้ 1 bite

long int → เก็บได้ 9 ล้านกว่า → 8 bite

ulong int → เก็บได้ หมื่นแปดล้าน

short int → 2 bite

ไม่ใส่ค่าให้พื้นที่เก็บเปลี่ยน

unsigned char → ทำให้เก็บข้อมูลที่ช่วง +

19 ธ.ค. 62

1e-2 → $1 \cdot 10^{-2}$
e = 10

- ขึ้นต้นด้วย 0 ตามด้วยตัวเลข คือ ฐาน 8
- ขึ้นต้นด้วย 0x คือ ฐาน 16
- ขึ้นต้นด้วย 0b, 0B คือ ฐาน 2

} ใช้เวลาเขีย
โค้ด

Ex.

```
int x = 0x13;
printf("%o (8)\n", x);   — แปลงเป็นฐาน 8
printf("%d (10)\n", x);  — แปลงเป็นฐาน 10
printf("%x (16)\n", x);  } แปลงเป็นฐาน 16
printf("%X (16)\n", x);
```

ทดลองผล

```
23 (8)
19 (10)
13 (16)
```

%c → พิมพ์อักขระ   Ex. printf("%c\n", x);
\n → new line

underflow → ลด, หายไป ตรงข้ามกับ overflow

~~[illegible]~~ char ตัวแปร = '\0'; → จะได้ string ว่าง

int หาร int คือ หารไม่เอาเศษ → 1/2 ได้ 0
1.0/2 ได้ 0.5

% (mod) ยังใช้ใน C ได้อยู่