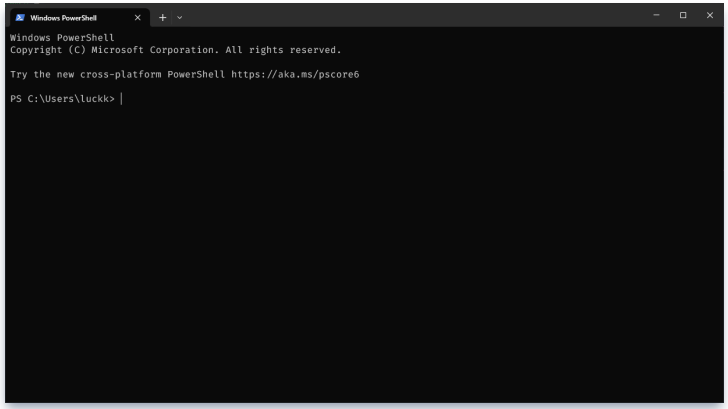
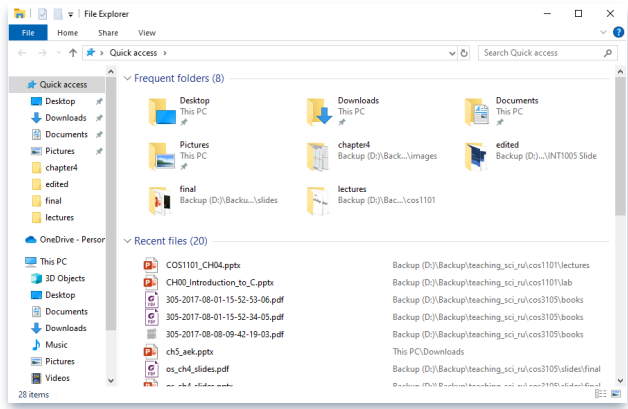
# Lab 1 - Basic C Programming Language

# รูปแบบของโปรแกรม




1. Command Line ✓




2. Graphical User Interface ✗

# โครงสร้างโปรแกรมภาษา C

1. Preprocessor Directive
2. Main Function
3. Statement

```c
#include <stdio.h> // 1. Preprocessor Directive

int main(){  // 2. Main Function
    printf("Hello World!"); // 3. Statement
    return 0; // Statement
}
```

# OUTPUT ??

# โครงสร้างโปรแกรมภาษา C

1. Preprocessor Directive
2. Main Function
3. Statement

```c
#include <stdio.h> // 1. Preprocessor Directive

int main(){  // 2. Main Function
    printf("Hello World!"); // 3. Statement
    return 0; // Statement
}
```

# OUTPUT ??

# โครงสร้างโปรแกรมภาษา C

1. Preprocessor Directive
2. Main Function
3. Statement

```c
#include <stdio.h> // 1. Preprocessor Directive

int main(){  // 2. Main Function
    printf("Hello World!"); // 3. Statement
    return 0; // Statement
}
```

# OUTPUT ??

# โครงสร้างโปรแกรมภาษา C

1. Preprocessor Directive
2. Main Function
3. Statement

```c
#include <stdio.h> // 1. Preprocessor Directive

int main(){  // 2. Main Function
    printf("Hello World!"); // 3. Statement
    return 0; // Statement
}
```

# OUTPUT ??

# โครงสร้างโปรแกรมภาษา C

1. Preprocessor Directive
2. Main Function
3. Statement

```c
#include <stdio.h> // 1. Preprocessor Directive

int main(){  // 2. Main Function
    printf("Hello World!"); // 3. Statement
    return 0; // Statement
}
```

# OUTPUT ??

```
Hello World
```

# Example

- https://onlinegdb.com/AD3bU_HX0
- 
- กดปุ่ม Fork

# โจทย์ 1

1. ลองเปลี่ยนจาก Hello World เป็น Hello Student!
   - เปลี่ยนเสร็จแล้วกดปุ่ม ▶ Run

```
#include <stdio.h>

int main(){
    printf("Hello Student!");
    return 0;
}
```

2. ลองเปลี่ยนจาก Hello World เป็นรหัสนักศึกษา
   - เปลี่ยนเสร็จแล้วกดปุ่ม ▶ Run

```
#include <stdio.h>

int main(){
    printf("46270304"); // Student Id
    return 0;
}
```

# ฟังก์ชันหลักของโปรแกรม (Main Function)

⚠ ทุกโปรแกรมต้องมี มี โดยโปรแกรมหลักจะเริ่มต้นด้วย main() และตามด้วยเครื่องหมายปีกกาเปิด `{` และปีกกาปิด `}`

```
int main()
{
    return 0;
}
```

- แต่ละคำสั่ง (Statement) จะต้องจบด้วยเซมิโคลอน `;` (Semicolon)
- ต้องมี return 0; เสมอ และต้องใส่เลขจำนวนเต็ม เช่น 0 = Success , 1 = Failure

```
return 0; // Success
return 1; // Failure
```

# ฟังก์ชั่นพิมพ์ข้อความบนหน้าจอ (printf)

- เป็นคำสั่งที่ใช้ในการแสดงผลออกทางจอภาพ โดยมีรูปแบบการใช้งานดังนี้
  - control หรือ format string เป็นส่วนที่ใส่ข้อความที่จะแสดงผล และส่วนควบคุมลักษณะการแสดงผล รวมทั้งบอกตำแหน่งที่ตัวแปรจะแสดงผล
  - variable list เป็นตัวแปรที่ต้องการจะแสดงผล ในกรณีที่ต้องการแสดงข้อความ ไม่จำเป็นต้องมีส่วนนี้

```
printf("ข้อความ หรือ control หรือ format string", variable list );
```

# Example 2

```c
#include <stdio.h>

int main(){
    printf("Hello Ramkamhaeng University");
    printf("Department of Computer Science");
    return 0;
}
```

# OUTPUT

```
Hello Ramkamhaeng UniversityDepartment of Computer Science
```

- จะให้ข้อความไม่ติดกันแบบนี้จะทำยังไง?

```
Hello Ramkamhaeng University
Department of Computer Science
```

# การใช้อักขระควบคุมการแสดงผล

- คำสั่ง `printf( )` สามารถควบคุมการแสดงผล ด้วยอักขระที่มี backslash นำหน้า

| | |
|---|---|
| \n | ขึ้นบรรทัดใหม่ |
| \t | เว้นระยะ 1 tab |
| \\ | แสดง \ |
| \" | แสดง " |

- CODE

```c
#include <stdio.h>
int main(){
    printf("Hello Ramkamhaeng University\n");
    printf("Department of Computer Science");
    return 0;
}
```

- OUTPUT

```
Hello Ramkamhaeng University
Department of Computer Science
```

# Example

- https://onlinegdb.com/AD3bU_HX0
- 
- กดปุ่ม Fork

# โจทย์ 2

- เขียนโปรแกรมให้แสดงข้อความดังนี้

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\  Hello, Krit Chomaitong  \\
\\  Ramkamhaeng University  \\
\\  Computer Science        \\
\\  "46270304"              \\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

# ชนิดของตัวแปร

- Characters
- Integers
- Floating points ( Real numbers )

# Characters

- ภาษา C จะรู้จักตัวอักษรอยู่ 256 ตัวอักษร ซึ่งจะเรียกลุ่มตัวอักษรนี้ว่า ASCII Table เช่น `'@'` `'a'` `'A'` `'4'` `'%'` `'+'` `'-'` `''`
- ถ้ามีตัวอักษรมากกว่า 1 ตัว แล้วอยู่ใน `""` จะเรียกว่า string เช่น `"Hello World"`

| Dec | Hex | Oct | Char | Description | Dec | Hex | Oct | Char | Description |
|---|---|---|---|---|---|---|---|---|---|
| 64 | 40 | 100 | @ | Commertial at/At sign | 69 | 45 | 105 | E | Latin capital letter E |
| 65 | 41 | 101 | A | Latin capital letter A | 70 | 46 | 106 | F | Latin capital letter F |
| 66 | 42 | 102 | B | Latin capital letter B | 71 | 47 | 107 | G | Latin capital letter G |
| 67 | 43 | 103 | C | Latin capital letter C | 72 | 48 | 108 | H | Latin capital letter H |
| 68 | 44 | 104 | D | Latin capital letter D | 73 | 49 | 109 | I | Latin capital letter I |

# Integers & Float

- คือตัวเลขฐาน 10 เช่น 10 54 0 -121 -68 752
- ถ้ามีจุดทศนิยม ก็จะเรียกว่า Float เช่น 547.43 0.0 0.44384 9.1923 -168.470 .22
- CODE

```c
#include <stdio.h>
int main(){
    printf("I am learning the %c programming language\n", 'C');
    printf("I knew %d data types in %c programming language\n", 3, 'C');
    printf("I am %.1f percent(%%) ready to move on ", 99.9);
    printf("to the next section!\n");
    return 0;
}
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

# Integers & Float

- คือตัวเลขฐาน 10 เช่น 10 54 0 -121 -68 752
- ถ้ามีจุดทศนิยม ก็จะเรียกว่า Float เช่น 547.43 0.0 0.44384 9.1923 -168.470 .22
- CODE

```c
#include <stdio.h>
int main(){
    printf("I am learning the %c programming language\n", 'C');
    printf("I knew %d data types in %c programming language\n", 3, 'C');
    printf("I am %.1f percent(%%) ready to move on ", 99.9);
    printf("to the next section!\n");
    return 0;
}
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
I am learning the C programming language
I knew 3 data types in C programming language
I am 99.9 percent(%) ready to move on to the next section!
```

# การ comment code

- การ comment code เป็นการอธิบายโปรแกรมว่าทำงานอย่างไร จะมีประโยชน์กับ Programmers คนอื่นๆ ที่มาแก้ไข Code ของเราต่อ ช่วยลดการหา Bugs

```
return ((s1 < s2) ? s1 : s2);
```

- แต่ถ้าเขียน comment กำกับด้วยจะเข้าใจง่ายขึ้น

```
return ((s1 < s2) ? s1 : s2); /* หาจำนวนที่น้อยที่สุดระหว่าง 2 จำนวน */
```

- ถ้าในกรณีนี้ไม่จำเป็น

```
printf("Payroll") /* Prints the word "Payroll" */
```

# การ comment code (2)

- เครื่องหมาย `/*` มีไว้สำหรับการ comment หลายๆบรรทัด `*/`

```
/*
บรรทัด 1
บรรทัด 2
*/
```

- เครื่องหมาย `//` มีไว้สำหรับการ comment บรรทัดเดียว

```
printf("No newline at the end!"); // ไม่มี \n ครั้งถัดไปที่ printf จะติดกัน
```

# การใช้ Whitespace

- ถ้าเขียน Code ติดกันจะอ่านได้ยาก ไม่แนะนำ

```
#include<stdio.h>
int main(){float a=0,b=1;printf("There are 2 numbers: %f and %f", a, b);return 0;}
```

- การใช้ Whitepsace หรือช่องว่างหรือการขึ้นบรรทัดใหม่ จะทำให้ Code อ่านได้ง่ายขึ้น

```
#include<stdio.h>

int main(){
    float a=0, b=1;

    printf("There are 2 numbers: %f and %f", a, b);

    return 0;
}
```

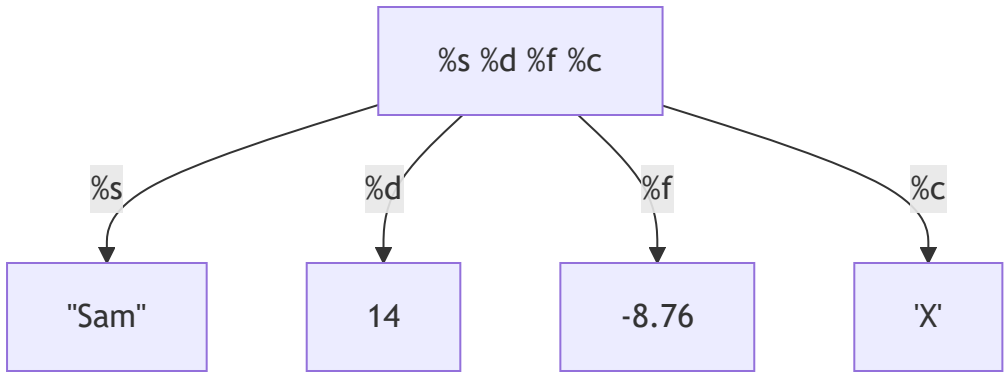
# การแปลงข้อมูลเป็น Characters เพื่อพิมพ์ทางหน้าจอ

| Conversion Character | Description |
|---|---|
| %d | Integer |
| %f | Floating-point |
| %c | Character |
| %s | String |

# การแปลงข้อมูลเป็น Characters เพื่อพิมพ์ทางหน้าจอ (2)

- ตัวอย่าง

```
printf("%s %d %f %c\n", "Sam", 14, -8.76, 'X');
```



# การแปลงข้อมูลเป็น Characters เพื่อพิมพ์ทางหน้าจอ (3)

- CODE

```
printf("%f %.3f %.2f, %.1f", 4.5678, 4.5678, 4.5678, 4.5678);
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

- CODE

```
printf("%5d", 12); //พิมพ์ให้ได้ 5 ตัว
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

- CODE

```
printf("%5.3f", 1.2); // พิมพ์ให้ได้ 5 ตัว โดยคิดจากจุดทศนิยมก่อน
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

# การแปลงข้อมูลเป็น Characters เพื่อพิมพ์ทางหน้าจอ (3)

- CODE

```
printf("%f %.3f %.2f, %.1f", 4.5678, 4.5678, 4.5678, 4.5678);
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
4.567800 4.568 4.57, 4.6
```

- CODE

```
printf("%5d", 12); //พิมพ์ให้ได้ 5 ตัว
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

- CODE

```
printf("%5.3f", 1.2); // พิมพ์ให้ได้ 5 ตัว โดยคิดจากจุดทศนิยมก่อน
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

# การแปลงข้อมูลเป็น Characters เพื่อพิมพ์ทางหน้าจอ (3)

- CODE

```
printf("%f %.3f %.2f, %.1f", 4.5678, 4.5678, 4.5678, 4.5678);
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
4.567800 4.568 4.57, 4.6
```

- CODE

```
printf("%5d", 12); //พิมพ์ให้ได้ 5 ตัว
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
   12 // มีอักขระว่าง 3 ตัวนำ ___12
```

- CODE

```
printf("%5.3f", 1.2); // พิมพ์ให้ได้ 5 ตัว โดยคิดจากจุดทศนิยมก่อน
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

# การแปลงข้อมูลเป็น Characters เพื่อพิมพ์ทางหน้าจอ (3)

- CODE

```
printf("%f %.3f %.2f, %.1f", 4.5678, 4.5678, 4.5678, 4.5678);
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
4.567800 4.568 4.57, 4.6
```

- CODE

```
printf("%5d", 12); //พิมพ์ให้ได้ 5 ตัว
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
   12 // มีอักขระว่าง 3 ตัวนำ ___12
```

- CODE

```
printf("%5.3f", 1.2); // พิมพ์ให้ได้ 5 ตัว โดยคิดจากจุดทศนิยมก่อน
```

- OUTPUT?? ให้นักศึกษาลองหาคำตอบ

```
1.200 // มีอักขระว่าง 3 ตัวนำ 1.200
```

# หลักการตั้งชื่อตัวแปร

- ต้องขึ้นต้นด้วยตัวอักษรภาษาอังกฤษ (ตัวใหญ่หรือเล็กก็ได้) หรือขีดล่าง `_`
- ตามด้วยตัวอักษรภาษาอังกฤษ ตัวเลข หรือขีดล่าง (Underscore) `_`
- ***ไม่มีช่องว่างหรือตัวอักษรพิเศษอื่นๆ*** เช่น `!` , `@` , `#` , `$` , `%` , `^` เป็นต้น
- ตัวพิมพ์ใหญ่และเล็กจะเป็นคนละตัวกันเช่น NAME, name, Name, NamE
- ห้ามซ้ำกับคำสงวน Reserved Words ของภาษา C เช่น char, do, const, break,...
- ห้ามตั้งชื่อซ้ำกับ Function ที่อยู่ใน Library ของภาษา C เช่น printf, scanf, ...
- ไม่เกิน 31 ตัวอักษร

# การประกาศตัวแปร

- ประกาศด้วย type ตามด้วย ชื่อตัวแปร

```
char answer;
int quantity;
float price;
int x,y;
```

- ประกาศค่าเริ่มต้นให้กับตัวแปร

```
answer = 'B';
quantity = 14;
price = 7.95;
x = 0;
y = 1;
```