C Programming Language

(2nd class)

Dohyung Kim

Assistant Professor @ Department of Computer Science

Today …

- Variables
- Output to a monitor
- Input from a keyboard
- Output/input (to/from a file)

Variables

- **Declaring, Initializing, and Assigning Variables**
  - Declaration

```
int num;
int num2, num3;
```

  - Initialization

```
int num = 3;
```

  - Assignment

```
num = 4;
num2 = num;
num3 = num2 = num1 = 5;
```

Variables (cont’d)

Basic Data Types

- int (4 byte-sized : 2^32 integer numbers) → int a = 30000;
- char (1 byte-sized : 2^8 characters) → char b = ‘a’
- float (4 byte-sized) → double c = ‘3.1415f’
- double (8 byte-sized) → double d = ‘3.1415’

Data type modifiers

- short(long) int a;
- Unsigned(signed) int b;

Variables (cont’d)

- **Naming variables**
  - Made up of letters (upper and lower case), digits, and under-score character (“_”)
  - Names must not begin with a digit
  - Keywords in the language cannot be used
  - Example
    - valid : foo, Bar, NUM, foo_bar, _foo, QuXu
    - Invalid : 2foo, my foo, $foo, while

Output to a monitor

```
#include <stdio.h>
int main (void)
{
    printf (“hello, world!\n”);
    return 0;
}
```

```
#include <stdio.h>
int main (void)
{
    int res;
    res = 10+20;
    printf (“10+20 = %d \n”, res);
    printf (“program ends”);
    return 0;
}
```

* Format specifier

%d – int

%ld – long int

%f – float

%lf – double

%c – char

%s – string

Output to a monitor

```
#include <stdio.h>
int main (void)
{
    int res;
    res = 10+20;
    printf(“10+20 = %d \n”, res);
    puts(“program ends”);
    // function ‘puts’ is used for only displaying a simple string not formatted one
    return 0;
}
```

Output Example

```
#include <stdio.h>
void myfunc (int a, int b, int c)
{
   printf (“%d%d%d”, a, b, c);
}

void main()
{
    int i=1, j=2, k=3;
    myfunc(i, j, k);
}
```

Input from a keyboard

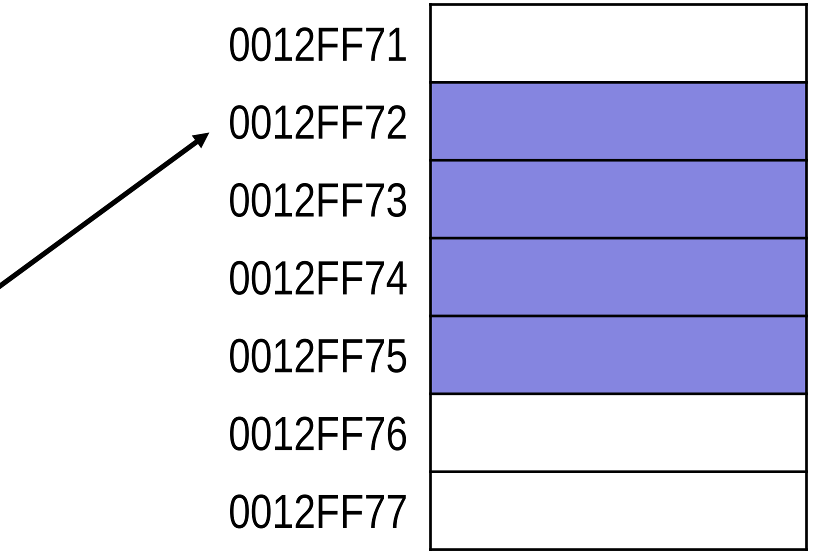
```
#include <stdio.h>
int main (void)
{
    int var;
    printf (“Please input an integer value :  ”);
    scanf (“%d”, &var);
    printf (“you entered : %d \n”, var);
    return 0;
}
```

Input from a keyboard

```
#include <stdio.h>
int main (void)
{
    int var;
    printf (“Please input an integer value :  ”);
    scanf (“%d”, &var);
    printf (“you entered : %d \n”, var);
    return 0;
}
```

Symbol table

name	type	address
var	int	0012FF72



Input from a keyboard

```
#include <stdio.h>
int main (void)
{
    int studentID, age;
    printf (“Please input your student ID and age:  ”);
    scanf (“%d %d”, &studentID, &age);
    printf (“your student ID : %d, age : %d\n”, studentID, age);
    return 0;
}
```

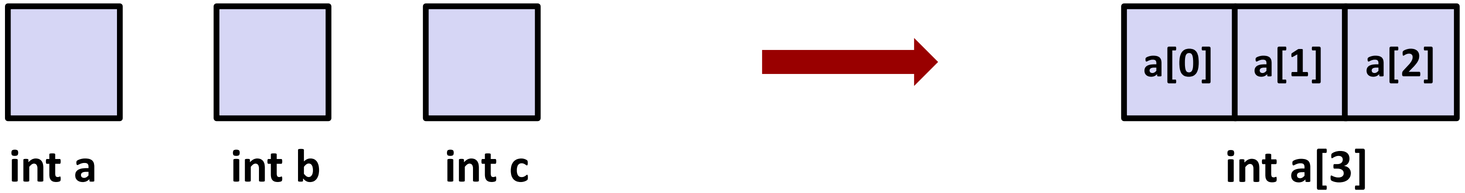
Input from a keyboard

```
#include <stdio.h>
int main (void)
{
    char name[20];
    printf (“Please your name :  ”);
    scanf (“%s”, name);
    printf (“you entered : %s \n”, name);
    return 0;
}
```

Input from a keyboard

```
#include <stdio.h>
int main (void)
{
    char name[20];
    printf (“Please your name :  ”);
    scanf (“%s”, name);
    printf (“you entered : %s \n”, name);
    return 0;
}
```

* What is array?



int a

int b

int c

a[0]

a[1]

a[2]

int a[3]

Array example

```
#include <stdio.h>
int main (void)
{
    int myVar[4];
    myVar[0] = 1;
    myVar[1] = 3;
    printf(“Please input an integer value : ”);
    scanf (“%d”, &myVar[2]);
    myVar[3] = myVar[0] + myVar[1] + myVar[2];
    printf (“Values in the array of myVar : %d, %d, %d, %d\n”,
                                        myVar[0], myVar[1], myVar[2], myVar[3]);
    return 0;
}
```

Why do we need an array?

- In this class, there are 80 students.
- We want to get the average and standard deviation of their exam scores

```
 int s1 = 70; int s2 = 90; int s3 = 100; …. ; int s80 = 10;
 double avg = (s1 + s2 + … + s80)/80;
 double std = (s1*s1 + s2*s2 + … + s80*s80)*(s1*s1 + s2*s2 + … + s80*s80)/80 –
avg*avg;
```

Why do we need an array?

- In this class, there are 80 students.
- We want to get the average and standard deviation of their exam scores

```
 int s1 = 70; int s2 = 90; int s3 = 100; …. ; int s80 = 10;
 double avg = (s1 + s2 + … + s80)/80;
 double std = (s1*s1 + s2*s2 + … + s80*s80)*(s1*s1 + s2*s2 + … + s80*s80)/80 –
avg*avg;
```

- What if we have 100000 students?

Why do we need an array?

- In this class, there are 80 students.
- We want to get the average and standard deviation of their exam scores

```
int s[80] = {70, 90, 100 …. , 10};
int i, sum=0, sqsum = 0;
for (i=0; i<80; ++i)
{
      sum = sum + s[i];
      sqsum = sqsum + s[i]*s[i];
}
double avg = sum / 80;
double std = sqsum*sqsum/80 – avg*avg;
```

Why do we need an array?

- In this class, there are 80 students.
- We want to get the average and standard deviation of their exam scores

```
int s[80] = {70, 90, 100 …. , 10};
int i, sum=0, sqsum = 0;
for (i=0; i<80; ++i)
{
      sum = sum + s[i];
      sqsum = sqsum + s[i]*s[i];
}
double avg = sum / 80;
double std = sqsum*sqsum/80 – avg*avg;
```

What if we have 100000 students?

```
for (i=0; i<100000; ++i){
```

Input from a keyboard

```
#include <stdio.h>
int main (void)
{
    char name[20];
    printf (“Please your name :  ”);
    gets (name);
    printf (“you entered : %s \n”, name);
    return 0;
}
```

Output to a file

```
#include <stdio.h>
int main (void)
{
    FILE* fp;
    char name[20];
    printf (“Please input your name:  ”);
    scanf (“%s”, name);
    printf (“you entered : %s \n”, name);
    fp = fopen(“tmpFile.txt”, “w”);
    fprintf(fp, “Your name is %s\n”, name);
    fclose(fp);
    return 0;
}
```

*** Basic modes for fopen**

r – open for reading

w – create for writing or truncate to zero length

a – append; open for writing at the end of file

Output to a file

```
#include <stdio.h>
int main (void)
{
    FILE* fp;
    char name[20];
    printf (“This is a test code”);
    fp = fopen(“tmpFile.txt”, “w”);
    fputs(“you can write strings in the file\n”, fp);
    fclose(fp);
    printf (“program ends\n”);
    return 0;
}
```

Input from a file

```
#include <stdio.h>
int main (void)
{
    FILE* fp;
    char name[20];
    fp = fopen(“tmpFile.txt”, “r”);
    fscanf(fp, “%s”, name);
    printf(“I read %s from the file \n”, name);
    fclose(fp);
    return 0;
}
```

Input from a file

```
#include <stdio.h>
int main (void)
{
    FILE* fp;
    char name[20];
    fp = fopen(“tmpFile.txt”, “r”);
    fgets(name, 20, fp);
    printf(“I read %s from the file \n”, name);
    fclose(fp);
    return 0;
}
```

What we have covered today

- Variables
- Input/output functions
- Array, For (statement)

Q and A