Basis and Practice in Programming
week 2
Type of Variables (1/2)

- Many types in C language
  - Integer, Character, Floating-point ...

- `int`  
  - Integer type variable  
  - 4 bytes of memory space

- `char`  
  - Character type variable  
  - 1 byte of memory space

- `float`  
  - Real number type variable  
  - 4 bytes of memory space
/* Week 2 practice 1
 * size of each variables */
#include <stdio.h>

int main(void)
{
    printf("int : %d\n", sizeof(int));
    printf("long : %d\n", sizeof(long));
    printf("char : %d\n", sizeof(char));
    printf("float : %d\n", sizeof(float));
    printf("double : %d\n", sizeof(double));

    return 0;
}
Declare Variables

• Declaration
  – Every variables must be declared before using
  – Usage : <keyword> <name of variable>

```c
#include <stdio.h>

int main(void)
{
    a = 10;
    printf("a = %d\n", a);
    return 0;
}
```
• Declaration
  – Every variable must be declared before using
  – Usage: `<keyword> <name of variable>`

/* Week 2 practice 2
 * Declare variables */
#include <stdio.h>

int main(void)
{
    a = 10;
    printf("a = %d\n", a);
    return 0;
}

/* Week 2 practice 3
 * Declare variables */
#include <stdio.h>

int main(void)
{
    int a;
    a = 10;
    printf("a = %d\n", a);
    return 0;
}
Name of Variables (1/2)

• Composed of Alphabet, number and under-bar(_)

• Capital letter is not equal to small letter
  – int a and int A are different variable
  – Int (x) int (o), float (o) Float (x)

• Start character of name of variable
  – Must not use numbers

• Special keyword and space character are not allowable
Name of Variables (2/2)

/* Week 2 practice 4
* Name of variables */
#include <stdio.h>

int main(void)
{
    int a;
    int A;
    int int_a; //correct
    //float int //wrong
    char c, d; //declaration of two variables at the same time

    return 0;
}
scanf() and printf() (1/3)

• Input/Output function
  – Declared in standard input output header file (stdio.h)

• Formatted scan function : `scanf`
  – Usage : `scanf("<input type>", &<variable>);`

• Formatted print function : `printf`
  – Usage : `printf("<output type>", <variable>);`

• Input/output type
  – %d : decimal number, %c : character
/* Week 2 practice 5  
 * Name of variables */

#include <stdio.h>
int main(void)
{
    int result;
    int val1, val2;
    char c = 'k';

    printf("First number : ");
    scanf("%d", &val1);
    printf("Second number : ");
    scanf("%d", &val2);

    result = val1 + val2;
    printf("%c
", c);
    printf("%d + %d = %d
", val1, val2, result);

    return 0;
}
```c
#include <stdio.h>
int main(void)
{
    int result;
    int val1, val2;

    scanf("%d %d", &val1, &val2); //separated by space
    result = val1 + val2;
    printf("%d + %d = %d \n", val1, val2, result);

    return 0;
}
```
Arithmetic Operation (1/2)

- **Addition**
  - Usage: result = a + b;
- **Subtraction**
  - Usage: result = a – b;
- **Multiply**
  - Usage: result = a * b;
- **Division**
  - Usage: result = a / b;
- **Modular**
  - Usage: result = a % b;

```c
#include <stdio.h>

int main(void)
{
    int a, b, result;
    a = 10, b = 3;

    result = a + b;
    printf("%d + %d = %dn", a, b, result);

    result = a - b;
    printf("%d - %d = %dn", a, b, result);

    result = a * b;
    printf("%d * %d = %dn", a, b, result);

    result = a / b;
    printf("%d / %d = %dn", a, b, result);

    result = a % b;
    printf("%d %% %d = %dn", a, b, result);

    return 0;
}
```
Arithmetic Operation (2/2)

```c
/* Week 2 practice 8
 * Arithmetic operation */
#include <stdio.h>

int main(void)
{
    int a, b;
    a = 10, b = 3;

    a += b; // same as a = a + b
    printf("%d\n", a);
    a -= b; // same as a = a - b
    printf("%d\n", a);
    a *= b; // same as a = a * b
    printf("%d\n", a);
    a /= b; // same as a = a / b
    printf("%d\n", a);
    a %= b; // same as a = a % b
    printf("%d\n", a);

    return 0;
}
```
Exercise (1/3)

- Compile
  - gcc <filename>.c -o <output>
  - $gcc practice1.c -o practice1

- Make output file
  - $./practice1 < input_ex1.txt > output_ex1.txt

- Diff command
  - You must check your output by sample output
  - $diff output_ex1.txt sample_output.txt
  - If any message isn’t printed, submit your code to I-campus
Exercise (2/3)

• Practice
  – Compile practice6.c file
  – `$gcc practice6.c –o practice6`
  – Download sample input file / output file
  – Make output file
  – Check your output file with sample output file through `diff` command
Exercise (3/3)

- Week2 exercise
  - Simple arithmetic operating program
  - Enter two numbers
  - Print each results of operation ‘+’, ‘-’, ‘*’, ‘/’ and ‘%’ at new line
  - All outputs must be **INTEGER** format
  - Due date: Today’s 11:59 PM
  - Submit to I-campus

Input

| 1 | 3 |

Output

| 4 |
| -2 |
| 3 |
| 0 |
| 1 |