

Summary of Lesson 4

switch() case..

- enumeration of if(a == b)
- :(colon) is necessary after case statement
- useful for converting numerical representation to another representation
- keyword **break** is always necessary otherwise execute the rest
- keyword **default** means **else** in switch, if the input value doesn't coincide all the cases, the statements which declared after **default** will be executed

while()

- if the content of the parenthesis is **true**, the content of following curly bracket will be executed repeatedly
- the execution repeats until the content of the parenthesis becomes **false**
- if there is a keyword **break** in the curly bracket, the repetition stops immediately. following statements in the curly bracket will be **not** executed
- if there is a keyword **continue** in the curly bracket, current loop starts again from the beginning of curly bracket
- you can nest while loop

for(;;)

- sophisticated, efficient form of while
- keyword **break** and **continue** are also valid in this loop
- you can declare three things in the parenthesis at the same time
 - **initialization**
 - normally used for initialization of index variable
 - **declaration of condition**
 - same as function of the parenthesis of while loop
 - **statement**, executed at the end of loop
 - normally increment index variable

lesson4_1.c

```
#include <stdio.h>
/* an example of switch~case */
int main(void)
{
    int num;
    printf("input a number from 0 to 6:");
    scanf("%d",&num);

    switch(num)
    {
        case 0:
```

```
        printf("Monday");
        break;
    case 1:
        printf("Tuesday");
        break;
    case 2:
        printf("Wednesday");
        break;
    case 3:
        printf("Thursday");
        break;
    case 4:
        printf("Friday");
        break;
    case 5:
        printf("Saturday");
        break;
    case 6:
        printf("Sunday");
        break;
    default:
        printf("wrong number");
}
return 0;
}
```

lesson4_2.c

```
#include <stdio.h>
/* infinite loop */
int main(void)
{
    while(1)
    {
        printf("Stop!\n"); // stop manually with red button
    }
    return 0;
}
```

lesson4_3.c

```
#include <stdio.h>
/* print random even numbers */
int main(void)
{
    int x;
    while(1)
    {
        x = rand();
```

```
        if(x % 2) // if odd number, go back to first line
        {
            continue;
        }
        printf("%d\n",x);
    }
    return 0;
}
```

lesson4_4.c

```
#include <stdio.h>
/* print random numbers until first multiple of 100 */
int main(void)
{
    int x;
    while(1)
    {
        x = rand();
        printf("%d\n",x);
        if(!(x % 100)) // if multiple of 100 comes, the loop finishes
        {
            break;
        }
    }
    return 0;
}
```

lesson4_5.c

```
#include <stdio.h>
/* finite loop */
int main(void)
{
    int x = 0;
    while(x < 20)
    {
        printf("x is now %d\n", x);
        x++;
    }
    return 0;
}
```

lesson4_6.c

```
#include <stdio.h>
/* for loop */
int main(void)
{
```

```
int x;
for(x = 0;x < 20;x++)
{
    printf("x is now %d\n", x);
}
return 0;
}
```

lesson4_7.c

```
#include <stdio.h>
/* nested loop */
int main(void)
{
    int x,y;
    for(x = 1;x <= 9;x++)
    {
        for( y = 1; y <= 9 ; y++)
        {
            printf("%d x %d = %d\n",x,y,x*y);
        }
    }
    return 0;
}
```

lesson4_8.c

```
#include <stdio.h>
/* power calculator */
int main(void)
{
    int base,power,tmp,i; // power calculation

    printf("base:");
    scanf("%d",&base);
    printf("power:");
    scanf("%d",&power);
    tmp = base;
    power = power-1;

    for(i = 0;i < power;i++)
    {
        tmp *= base;
    }
    printf("result:%d",tmp);
    return 0;
}
```

lesson4_9.c

```
#include <stdio.h>
/* how a software works */
int main(void)
{
    short flag = 1;
    int x, y, result;
    char op, quit, dummy;
    while(flag)
    {
        printf("calc:");
        scanf("%d %c %d",&x, &op, &y);
        switch(op)
        {
            case '+':
                result = x + y;
                break;
            case '-':
                result = x - y;
                break;
            case '*':
                result = x * y;
                break;
            case '/':
                result = x / y;
                break;
        }
        printf("= %d\n",result);
        printf("quit?(y/n):");
        scanf("%c",&dummy);
        scanf("%c",&quit);
        if(quit == 'y')
        {
            flag = 0;
        }
    }
    puts("thank you!");
    return 0;
}
```