

Lecture 7.2

Topics

1. Statements – Brief
2. Conditional Statements/Structures – **if, if-else**

1. Statements – Brief

There are six types of statements in C and they are

- Expression Statement
- Compound (Block) Statement
- Labeled Statement
- Selection/Conditional Statement
- Iterative (Loop) Statement
- Jump Statement

The last statement, **JUMP STATEMENT**, is available but is discouraged to be used. Logic should be revised to avoid any jump or broken execution flow; thus to avoid the jump statement if possible.

We will be working with the other five statements during the course of the semester.

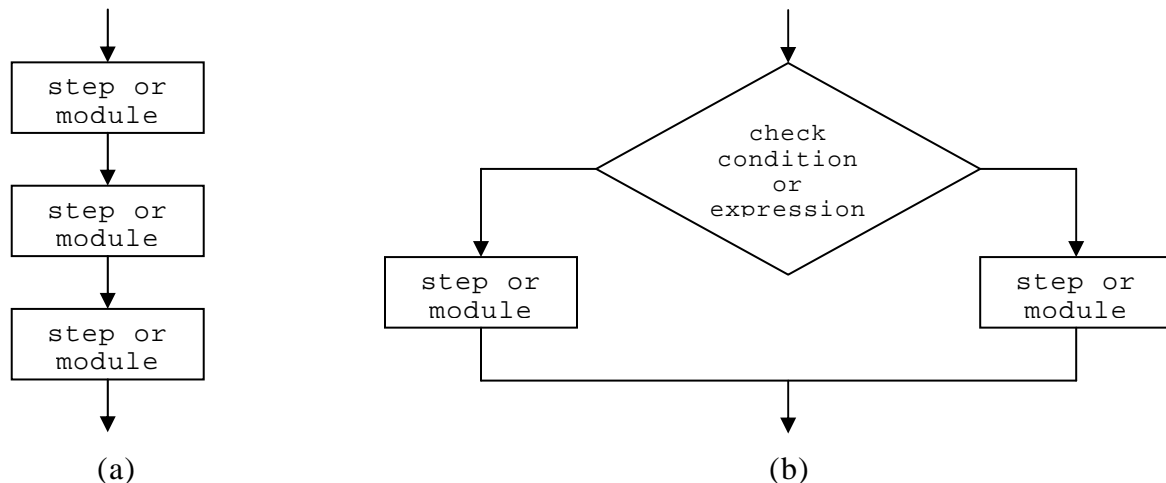
We have used the expression statements in many examples given in class. We will be working with the **conditional** and **loop** statements (also called structures). The compound and label statements will be parts of many of functions, conditional and loop statements.

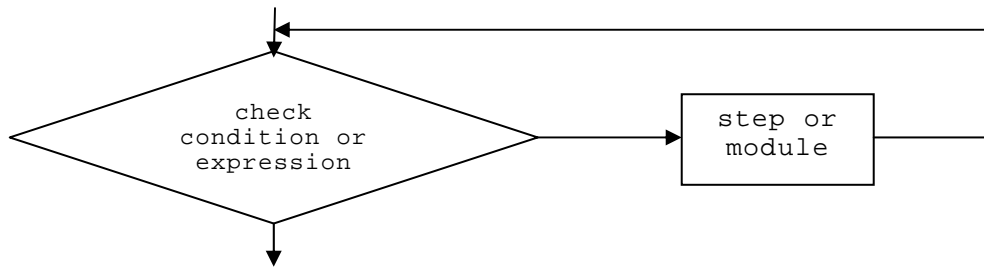
1.1 Basic Computer Programming Structures

In general, a programming structure is a basic unit of programming logic. There are three basic programming structures:

- (a) Sequence (or step),
- (b) Selection/decision, and
- (c) Repetition/loop.

A program may combine these three structures to produce the solution logic for a particular problem.





(c)

Figure 1 Three basic programming structures

1.2 Sequence (Step) Structure

A sequence structure will provide programming operations or events in sequence one after another or step by step.

This is depicted in **Figure 1a**, where each step or module can be just a single operation or several operations combined.

1.3 Selection/Decision or Conditional Structure

A selection or decision structure is depicted in **Figure 1b**. There is a set of expressions to be verified before the next operation can be followed.

In the basic structure, there are only two options to be considered as the outcomes of a decision. If the outcome represents a **true** then the flow will continue with one path. If the outcome represents a **false** then the flow will follow the other alternative path.

1.4 Loop/Repetition Structure

The loop structure is depicted in **Figure 1c** where the decision is checked before a selected event can be performed or followed.

There are different variations of loop structure. We will revisit them in later lectures.

2. Conditional Statements/Structures – **if, if-else**

There are several conditional structures such as **if**, **if-else**, **if-else-if**, **switch**. Two of these structures are introduced here:

(1) **if**, and

(2) **if-else**

2.1 Simple **if** Structure -- Flowchart

Syntax of **if** Structure

```

if ( testExpression ) {
    /*if testExpression is true, do something here.*/
}
  
```

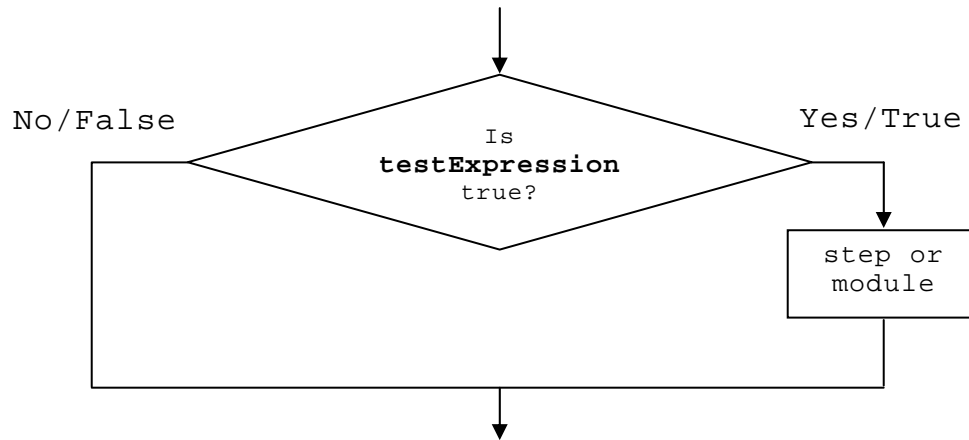


Figure 2 Conditional structure with one option

Example – Conditional **if** structure

```

/**
 *Program Name:   cis26L0721.c
 *Discussion:     if Structure
 */

#include <stdio.h>

/*Function prototypes*/
void printPositive( int );
void getPrintPositive( void );

/*Application driver*/
int main() {
    int iA;

    printf( "1234567890123456789012345678901234567890\n" );

    printf( "\nEnter an integer value: " );
    scanf( "%d", &iA );

    printPositive( iA );

    getPrintPositive( );

    printf( "\n" );
    return 0;
}

/*Function definitions*/
/**
 *Function Name: printPositive()
 *Description  : Printing information for an integer
 *Pre          : Integer to be assessed and printed
 *Post         : None
 */
void printPositive( int iOld ) {
    if ( iOld > 0 ) {
        printf( "\n%d is positive.\n", iOld );
    }
}

```

```

    return;
}

/**
 *Function Name: getPrintPositive()
 *Description  : Printing information for an integer
 *Pre          : Nothing
 *Post         : None
 */
void getPrintPositive( void ) {
    int iA;

    printf( "\nEnter an integer value: " );
    scanf( "%d", &iA );

    if ( iA > 0 ) {
        printf( "\n%d is positive.\n", iA );
    }

    return;
}

```

OUTPUT – Run #1

1234567890123456789012345678901234567890

Enter an integer value: 8

8 is positive.

Enter an integer value: 9

9 is positive.

OUTPUT – Run #2

1234567890123456789012345678901234567890

Enter an integer value: -8

Enter an integer value: 9

9 is positive.

OUTPUT – Run #3

1234567890123456789012345678901234567890

Enter an integer value: -8

Enter an integer value: -9

In the above flowchart, if the **testExpression** is **false** then no action would take place. However, it is possible to have different action for this (false) option with respect to the other (true) case.

Let's look at this conditional structure next.

2.2 if-else Flowchart

Syntax of if-else Structure

```

if ( testExpression ) {
    /*if testExpression is true, do something here.*/
} else {
    /*if testExpression is false, do something else.*/
}

```

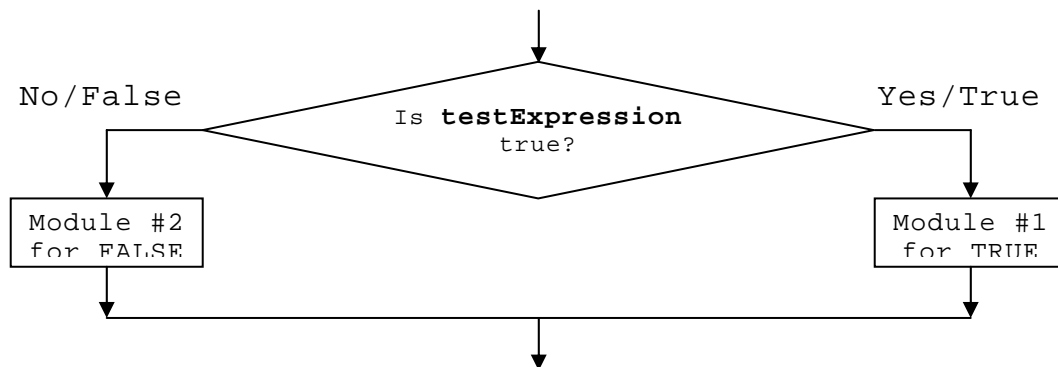


Figure 3 if-else conditional structure

Let's look at a revised version of the above functions.

```

/**
 *Function Name: printPositiveNegative()
 *Description  : Printing information for an integer
 *Pre          : Integer to be assessed and printed
 *Post         : None
 */
void printPositiveNegative( int iOld ) {
    if ( iOld > 0 ) {
        printf( "\n%d is positive.\n", iOld );
    } else {
        printf( "\n%d is non-positive.\n", iOld );
    }

    return;
}

/**
 *Function Name: getPrintPositive()
 *Description  : Printing information for an integer
 *Pre          : Nothing
 *Post         : None
 */
void getPrintPositive( void ) {
    int iA;

    printf( "\nEnter an integer value: " );
    scanf( "%d", &iA );

    if ( iA > 0 ) {
        printf( "\n%d is positive.\n", iA );
    } else {

```

```
    printf( "\n%d is non-positive.\n", iOld );  
}  
return;  
}
```