Basis and Practice in Programming

Chapter 6: “Control statements: Looping; Branching;”

Prof. Tamer ABUHMED
College of Software

성균관대학교
SUNG KYUN KWAN UNIVERSITY(SKKU)

So Far …

- We studied
    - Variable types, type casting, arithmetic and several logical operators
- We also studied
    - Strings, and basic integer and float arrays
- Besides, we studied
    - Looping (while)
- Now, we will study
    - Looping (for)
    - How to make decisions in your program (change the flow of your program)
    - We will study (if)

Lecture Objectives

- Explain the operation of the for loop
- Explain the logical operators
- Explain the multiple loops
- Explain the use of the “if statement”
- Explain the use of the “switch statement”

The For Loop

- The for loop
  - It gathers the three actions
    - Initialization (done once at the beginning of the first iteration)
      - ii = 0;
    - Condition (checked at the beginning of each iteration)
      - ii <= N;
    - Updating (done at the end of each iteration)
      - ii++

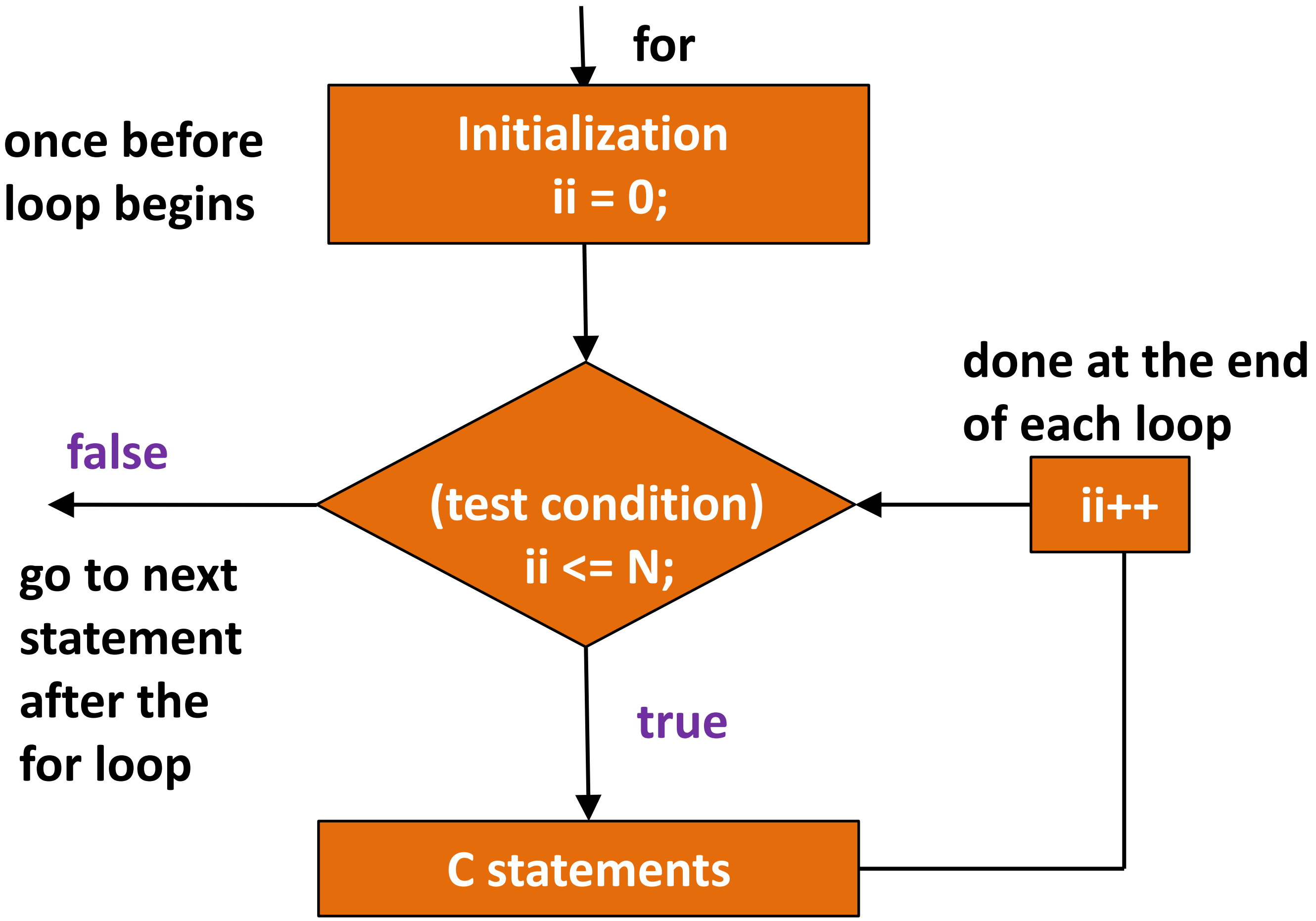
```
# include <stdio.h>
# define N 4

void main(void)
{
    int sum = 0, ii = 0;

    for(ii = 0; ii <= N; ii++)
    {
        sum += ii*ii;
    }
    printf(“The result is %d.\n”, sum);
}
```

Output

```
The result is 30.
```

The For Loop – contd.

- Detailed explanation of the for loop



The For Loop – contd.

- More examples

```
# include <stdio.h>
void main(void)
{
    char ch;
    for(ch = ‘a’; ch <= ‘z’; ch++)
        printf("The ASCII value for %c is %d.\n", ch, ch);

}
```

```
// sum the odd numbers from 1 to N
# include <stdio.h>
# define N 8

void main(void)
{
    int sum = 0, ii = 0;

    for(ii = 1; ii <= N; ii += 2)
    {
        sum += ii;
    }
    printf("result is %d.\n", sum);
}
```

Output

```
The result is 16.
```

The For Loop – contd.

- More examples

```
// sum the odd numbers from N to 1
# include <stdio.h>
# define N 7

void main(void)
{
    int sum = 0, ii = 0;

    for(ii = N; ii >= 1; ii -= 2)
    {
        sum += ii;
    }
    printf("result is %d.\n", sum);
}
```

Output

```
The result is 16.
```

The For Loop – contd.

- What does the following code do?
  - Multiple statements in one action separated by a comma

```
# include <stdio.h>
# define N 7

void main(void)
{
    int ii = 0, jj = 0;
    int result[N];

    for(ii = N-1, jj = 0; jj < N; ii--, jj++)
    {
        result[jj] = ii + jj;
    }

    for(ii = 0; ii < N; ii++)
        printf(“result[%d] = %d\n”, ii, result[ii]);

}
```

Output

```
result[0] = 6
result[1] = 6
result[2] = 6
result[3] = 6
result[4] = 6
result[5] = 6
result[6] = 6
```

The For Loop – contd.

- Can we omit one of the three actions?
  - Yes, but don’t omit the semicolon!

```c
# include <stdio.h>
# define N 7

void main(void)
{
    int ii = (N – 1), jj = 0;
    int result[N];

    for( ; jj < N; ii--, jj++)
    {
        result[jj] = ii + jj;
    }

    for(ii = 0; ii < N; ii++)
        printf(“result[%d] = %d\n”, ii, result[ii]);

}
```

Output

```
result[0] = 6
result[1] = 6
result[2] = 6
result[3] = 6
result[4] = 6
result[5] = 6
result[6] = 6
```

The For Loop – contd.

- Multiple loops
  - Multiplication tables

```c
# include <stdio.h>
# define N 5

void main(void)
{
    int ii = 0, jj = 0;

    for(ii = 0; ii < N; ii++)
    {
        for(jj = 0; jj < N; jj++)
        {
            printf("%d \t", (ii * jj));
        }
        printf("\n");
     }

}
```

Output

```
0         0         0         0         0
0         1         2         3         4
0         2         4         6         8
0         3         6         9         12
0         4         8         12        16
```

The if Statement

- The minimum of two numbers
  - It can be obtained using the logical operators we already studied

```
int num1 = 5, num2 = 7, min;

(num1 <= num2) ? (min = num1) : (min = num2);
// The following line is equivalent to the above line
// min = (num1 < num2) ? num1 : num2;
```

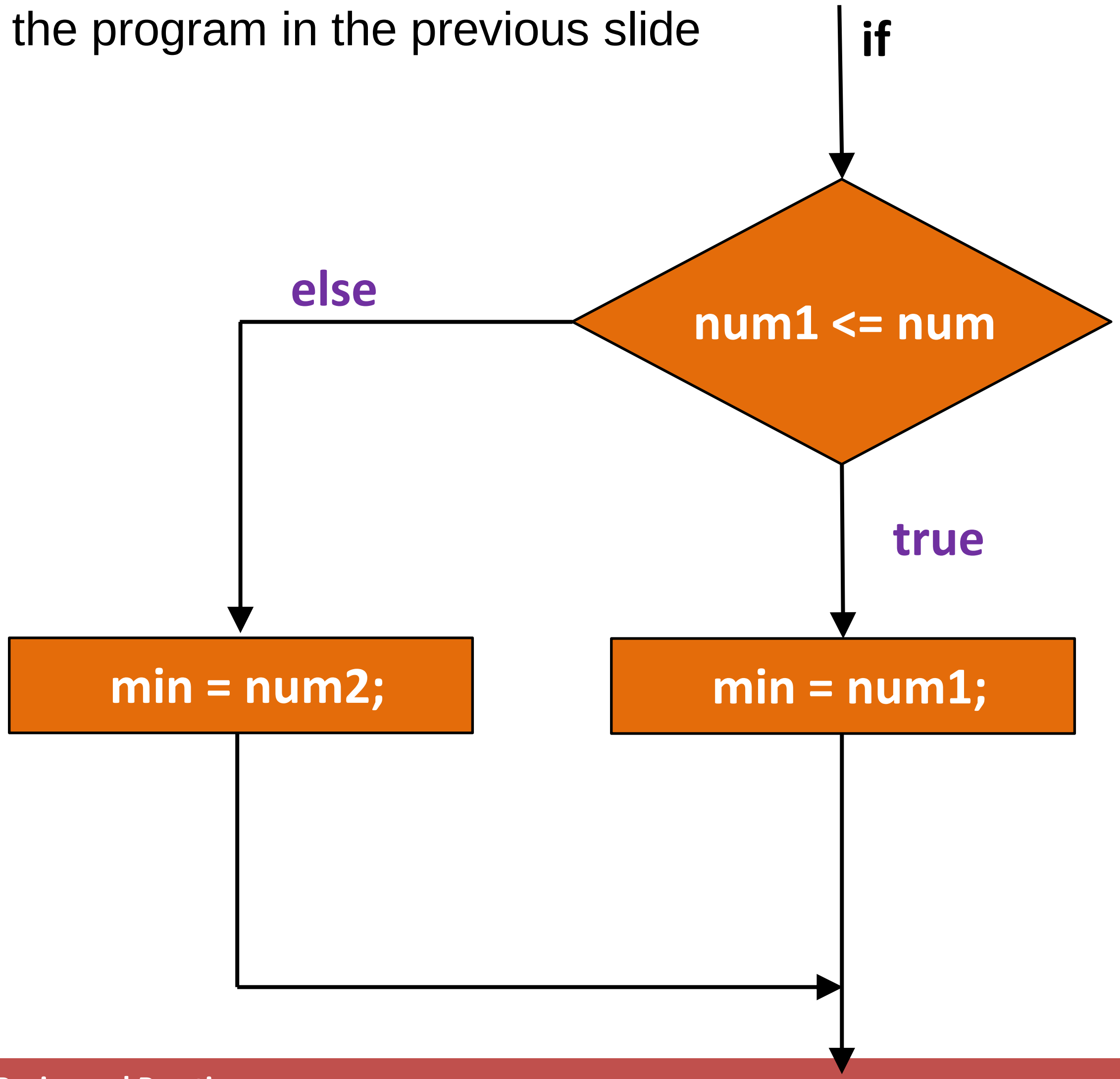
  - Also, it can be done using the “**if statement**”
    - 2 conditions

```
int num1 = 5, num2 = 7, min;

if(num1 <= num2){
    min = num1;
}
else{
    min = num2;
}

printf(“min = %d\n”, min);
```

```
int num1 = 5, num2 = 7, min;

if(num1 <= num2){
    min = num1;
}
else if(num1 > num2){
    min = num2;
}

printf(“min = %d\n”, min);
```

The if Statement – contd.

- The minimum of two numbers
  - Flow of the program in the previous slide



Logical and Arithmetic Operators - review

- Logical and arithmetic operators

Logical operators	
Operators	**meaning**
<, >	Less than, greater than
<=, >=	Less than or equal, greater than or equal
==	Equal
!=	Not equal

Arithmetic operators	
Operators	**meaning**
=	Equal (assignment)
+, -, /, *	Addition, subtraction, multiplication, division
%	Modulus
+=, -=, *=, /=, %=	x += y → x = x + y

The if Statement – contd.

- An “if statement” without else
  - We can use the “if statement” without any “else statement”
    - We have only one condition to check

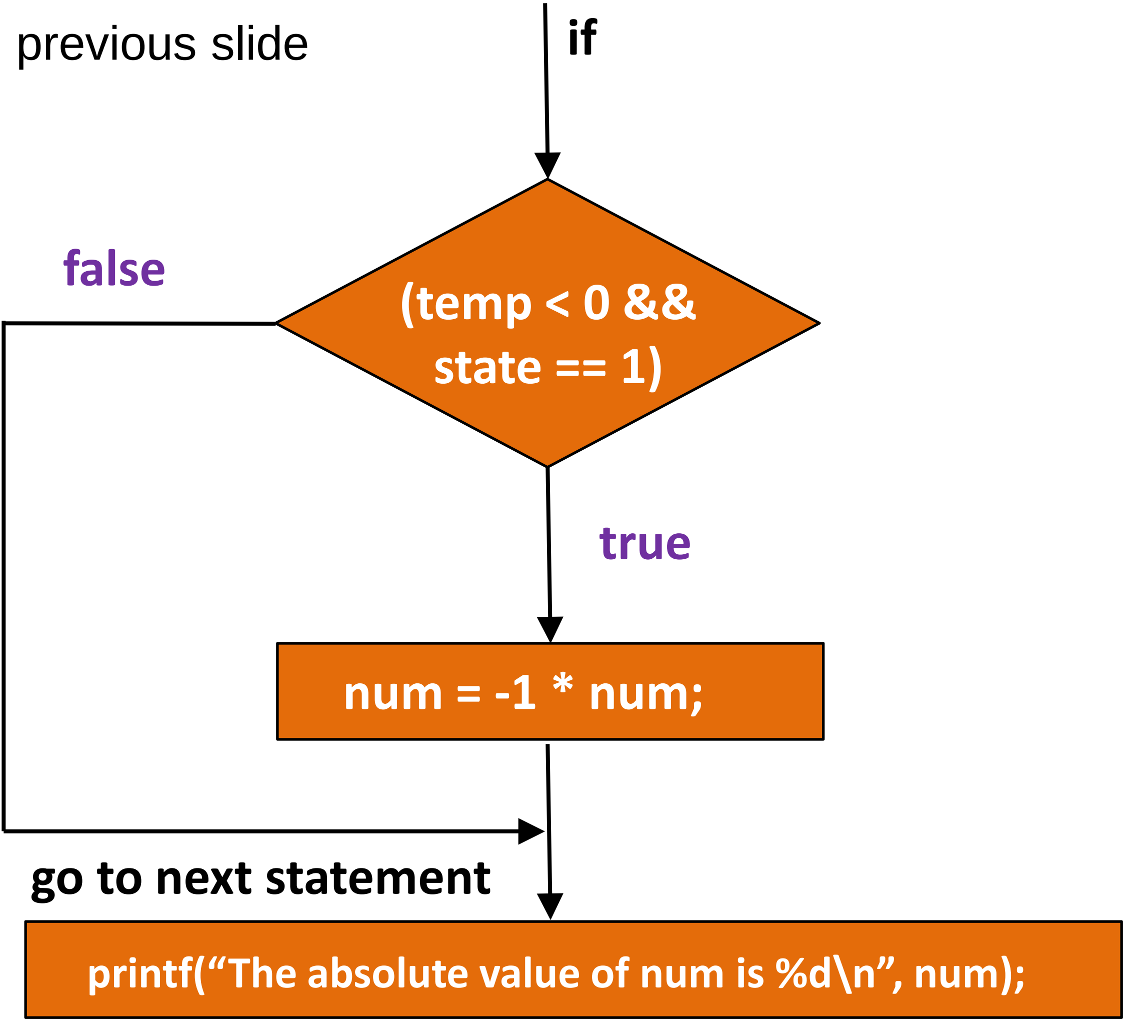
```
// find the absolute value of a number
# include <stdio.h>

void main()
{
     int num = 0, state;

     printf(“Please enter an integer: “);
     state = scanf(“%d”, &num);

     if(num < 0 && state == 1)
         num = -1 * num;

     printf(“The absolute value of num is %d\n”, num);

}
```

Output

```
Please enter an integer: -5
The absolute value of num is 5
```

The if Statement – contd.

- An "if statement" without else
  - Flow of program in previous slide



The if Statement – contd.

- A simple program if statement
  - A control panel (multiple conditions; 4 in this case)

```
# include <stdio.h>

# define low 5    // lower limit
# define high 28  // upper limit

void main()
{
     int temp = 0, state;

     printf(“Please enter the temperature: “);
     state = scanf(“%d”, &temp);

     if(temp <= low && state == 1)
         printf(“I'm freezing! Turn the heater on.\n”);
     else if(temp >= high && state == 1)
         printf(“It's too hot! Turn the air conditioner on.\n”); // Line 2
     else if( (temp < high) && temp > low && state == 1)
         printf(“The weather is amazing! Let's go party :)\n”);  // Line 3
     else
         printf(“Input error! Program is terminated.\n”)         // Line 4

}
```

Output 3

```
Please enter the temperature: 40
It's too hot! Turn the air conditioner on.
```

Output 1

```
Please enter the temperature: 4
I’m freezing! Turn the heater on.
```

Output 2

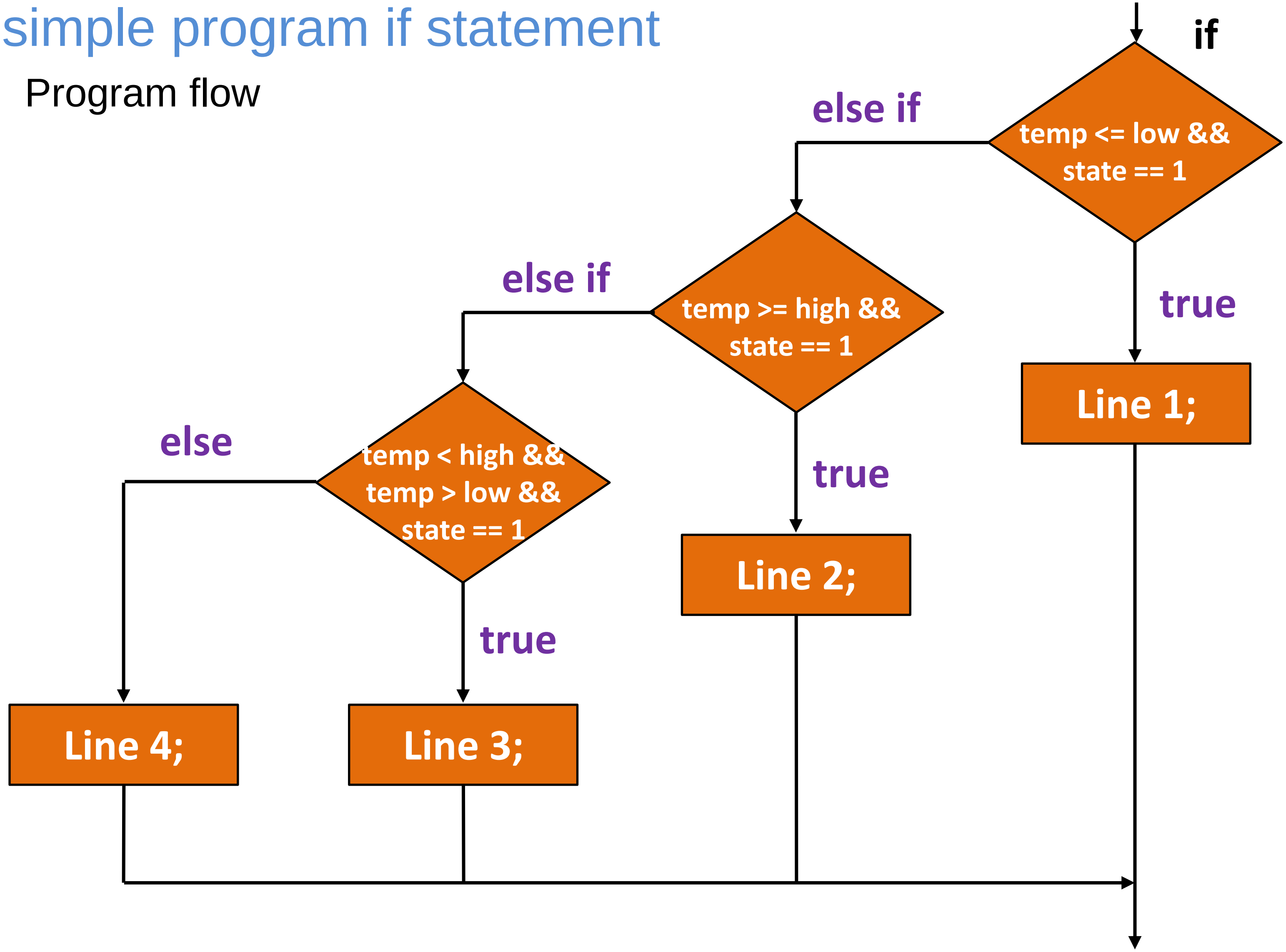
```
Please enter the temperature: 20
The weather is amazing! Let’s go party :)
```

Output 4

```
Please enter the temperature: q
Input error! Program is terminated.
```

The if Statement – contd.

- A simple program if statement
  - Program flow


if
temp <= low && state == 1
true
Line 1;
else if
temp >= high && state == 1
true
Line 2;
else if
temp < high && temp > low && state == 1
true
Line 3;
else
Line 4;

Example – multiple choices

```c
/* Program to evaluate simple expressions of the form
number operator number */
#include <stdio.h>
int main (void) {
     float value1, value2;
     char operator;
     printf ("Type in your expression.\n");
     scanf ("%f %c %f", &value1, &operator, &value2);
     if ( operator == '+' )
          printf ("%.2f\n", value1 + value2);
     else if ( operator == '-' )
          printf ("%.2f\n", value1 - value2);
     else if ( operator == '*' )
          printf ("%.2f\n", value1 * value2);
     else if ( operator == '/' )
          printf ("%.2f\n", value1 / value2);
     else printf ("Unknown operator.\n");
     return 0;
}
```

Output 1

```
Type in your expression.
4*8
32.00
```

Output 2

```
Type in your expression.
4+8
12.00
```

Output 2

```
Type in your expression.
8/4
2.00
```

Example - switch

```c
/* Program to evaluate simple expressions of the form
value operator value */
#include <stdio.h>
int main (void) {
     float value1, value2;
     char operator;
     printf ("Type in your expression.\n");
     scanf ("%f %c %f", &value1, &operator, &value2);
     switch (operator)  {
         case '+’:
         printf ("%.2f\n", value1 + value2);
         break;
         case '-': printf ("%.2f\n", value1 - value2); break;
         case '*': printf ("%.2f\n", value1 * value2); break;
         case '/':
            if ( value2 == 0 ) printf ("Division by zero.\n");
            else  printf ("%.2f\n", value1 / value2);
         break;
         default:  printf ("Unknown operator.\n");  break;
     }
     return 0;
}
```

The `switch` statement

```
switch ( expression )
{
     case value1:
          program statement
          program statement
          ...
          break;
     case value2:
          program statement
          program statement
          ...
          break;
          ...
     case valuen:
          program statement
          program statement
          ...
          break;
     default:
          program statement
          program statement
          ...
          break;
}
```

The expression is successively compared against the values value1, value2, ..., valuen. If a case is found whose value is equal to the value of expression, the program statements that follow the case are executed.

The switch test expression must be one with an integer value (including type char) (No float !).

The case values must be integer-type constants or integer constant expressions (You can't use a variable for a case label !)

The if Statement – contd.

- “if statement” with the “while loop”
  - This code converts small letter to CAPITAL letters

```
# include <stdio.h>

void main()
{
    char ch;

    ch = getchar();

    while(ch != '\n')
    {
        if(ch >= 97 && ch <= 122)  // small letter
            putchar(ch - 32);      // convert to capital letter
        else
            putchar(ch);           // print the ch as it is

        ch = getchar();            // get next character
    }

    putchar(ch);                   // print the new line

}
```

Output

```
HuEy @_@ FreeMan
HUEY @_@ FREEMAN
```

The if Statement – contd.

- Nested “if statements”
  - Check if a number is positive and divisible by 2

```c
# include <stdio.h>

void main()
{
    int num;

    printf("Please enter an integer: ");
    scanf("%d", &num);

    if(num > 0){  // first if
         if(num % 2 == 0){  // second if
             printf("The number is positive and even.\n");
         }
         else{  // else for the second if
             printf("The number is positive and odd.\n");
         }
    }
    else{     // else for the first if
         printf("The number is negative.\n");
    }
}
```

Output 1

```
Please enter an integer: 5
The number is positive and odd.
```

Output 2

```
Please enter an integer: 10
The number is positive and even.
```

Output 3

```
Please enter an integer: -5
The number is negative.
```

The conditional operator

condition ? *expression1* : *expression2*

```
maxValue = ( a > b ) ? a : b;
```

Equivalent to:

```
if ( a > b )
        maxValue = a;
else
        maxValue = b;
```

Example: conditional operator

```c
#include <stdio.h>

int main()
{
          int mark;
          printf("Enter mark: ");
          scanf("%d", &mark);
          printf("%s", mark >= 40 ? "Passed" : "Failed");
          return 0;
}
```

Output 1

```
Enter mark: 50
Passed
```

Output 2

```
Enter mark: 35
Failed
```

```c
#include <stdio.h>

int main()
{
          int n1 = 5, n2 = 10, max;

          // Largest among n1 and n2
          max = (n1 > n2) ? n1 : n2;
          printf("Largest number between %d and %d is %d. ",
                    n1, n2, max);
          return 0;
}
```

Output

```
Largest number between 5 and 10 is 10.
```

Lecture Keywords

- Keywords
  - The for loop
  - The multiple loops
  - The (if, else) condition statement
  - switch condition statement
  - conditional operator

Lecture Summary

- Explained the operation of the for loop
- Explained the logical operators
- Explained the multiple loops
- Explained the condition statement (if, else, switch)