

Tutorial

Programming & Data Structure: CS 11001

Section - 4/D

Department of Computer Science and
Engineering

I.I.T. Kharagpur

Spring Semester: 2013 - 2014 (16.01.2014)

Download

**Download the file tut160114.pdf from
Programming & Data Structures ... of**

<http://cse.iitkgp.ac.in/~goutam>

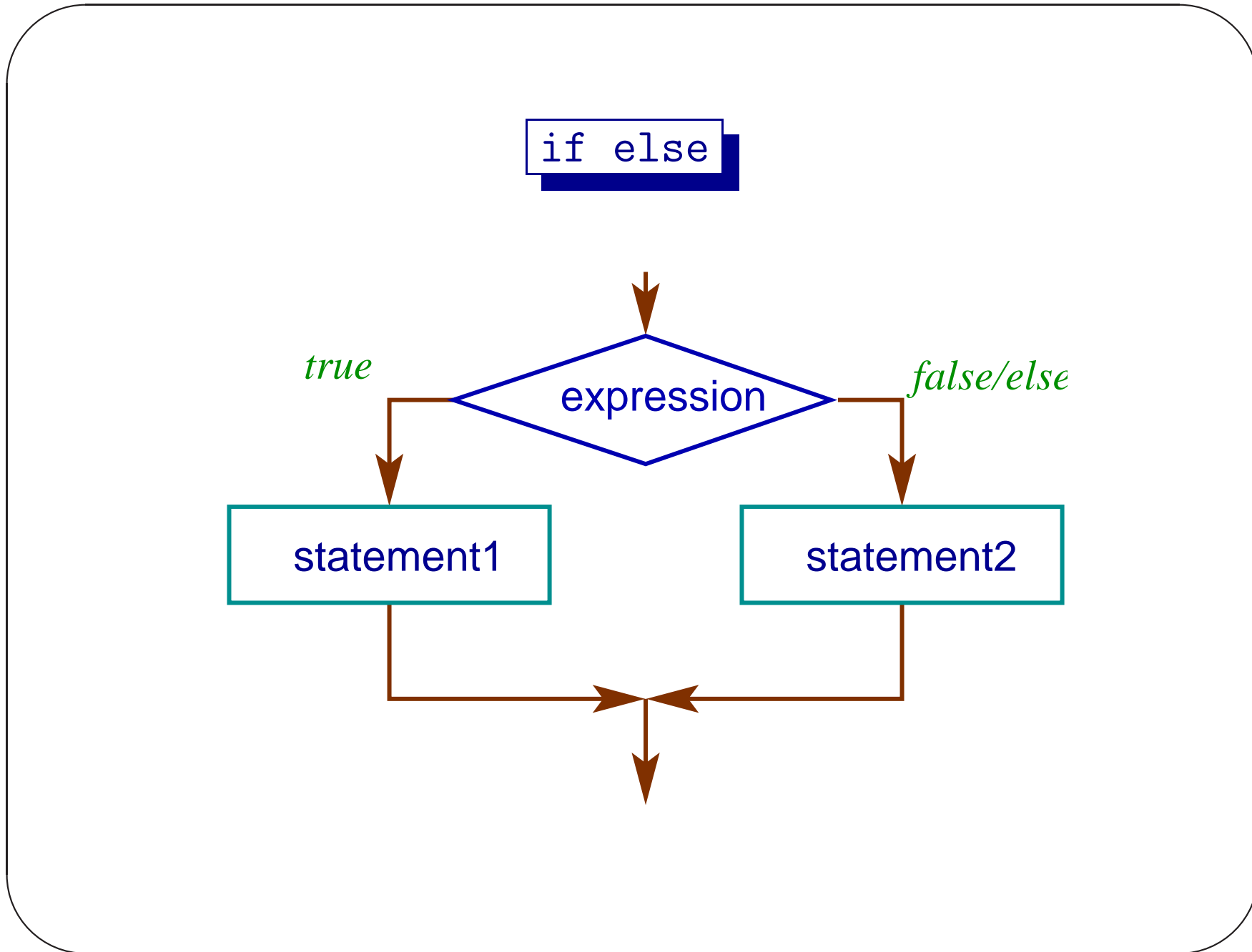
**View the file using the command `acroread` & or
`xpdf` &**

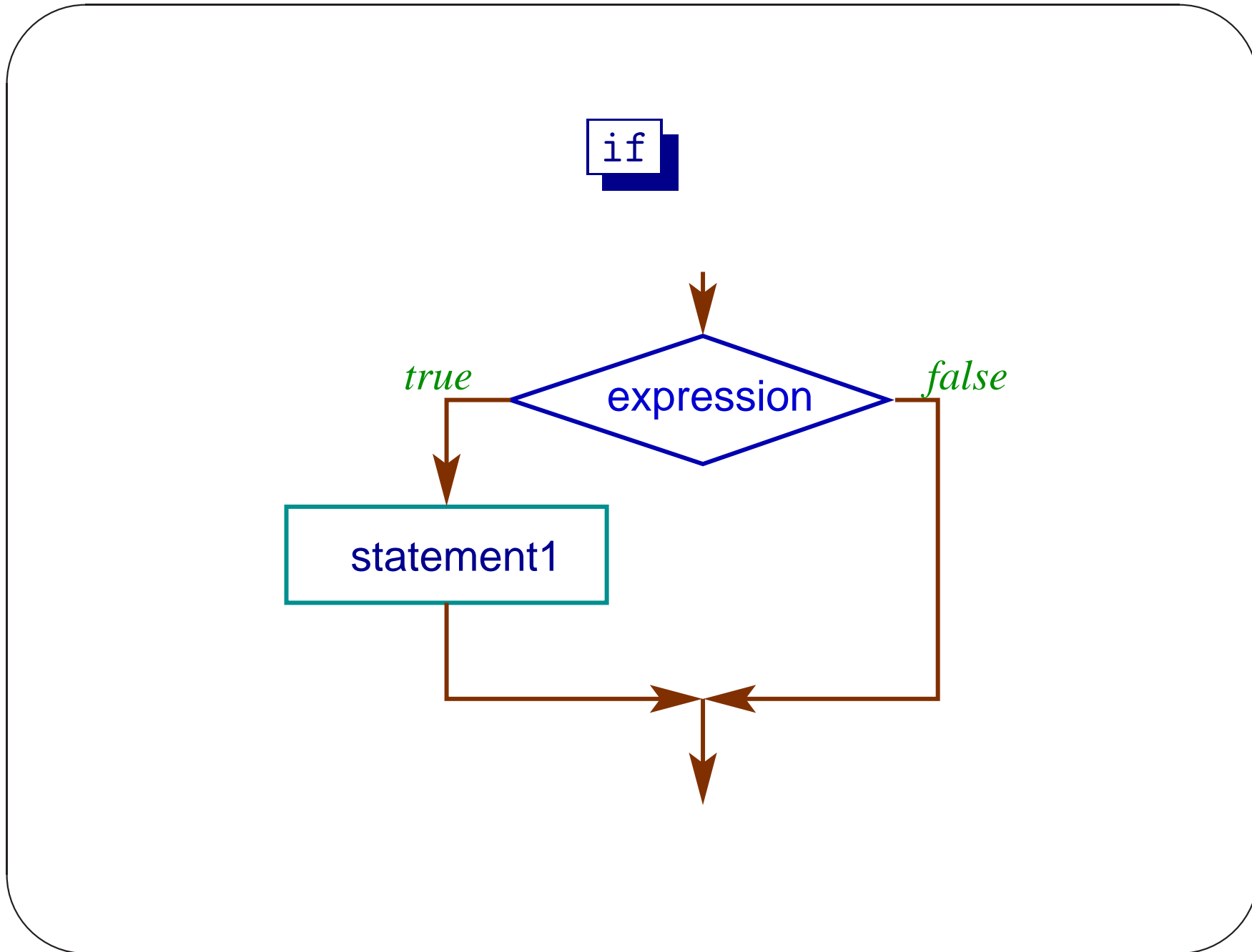
if Statement

We use **if-statement** for controlling the execution sequence in a program. The structure or syntax of **if**-statements are as follows.

```
if (expression) statement1 else statement2
```

```
if (expression) statement1
```





Compound Statement

A sequence of statements within a pair of curly braces $\{ \}$ forms a compound statement or block.

$$\{ \textit{statement}_1 \cdots \textit{statement}_k \}$$

```
#include <stdio.h>
int main() // temp18.c
{
    int data1, data2, larger;
    printf("Enter two integer data: ");
    scanf("%d%d", &data1, &data2);
    if(data1 > data2) larger = data1 ;
    else larger = data2;
    printf("\n%d is the larger among %d & %d\n",
           larger, data1, data2);
    return 0 ;
}
```

Tutorial II.1

Modify the C code of `temp18.c` so that exactly the same output will be printed for the same input, but the modified code will not use the variable `larger`.

Tutorial II.2

Write a C program that will read three integer data and will print the largest among them. It is not necessary to print input data.

Tutorial II.3

Write a C program to solve the previous problem (Tutorial II.2) using only two variables.

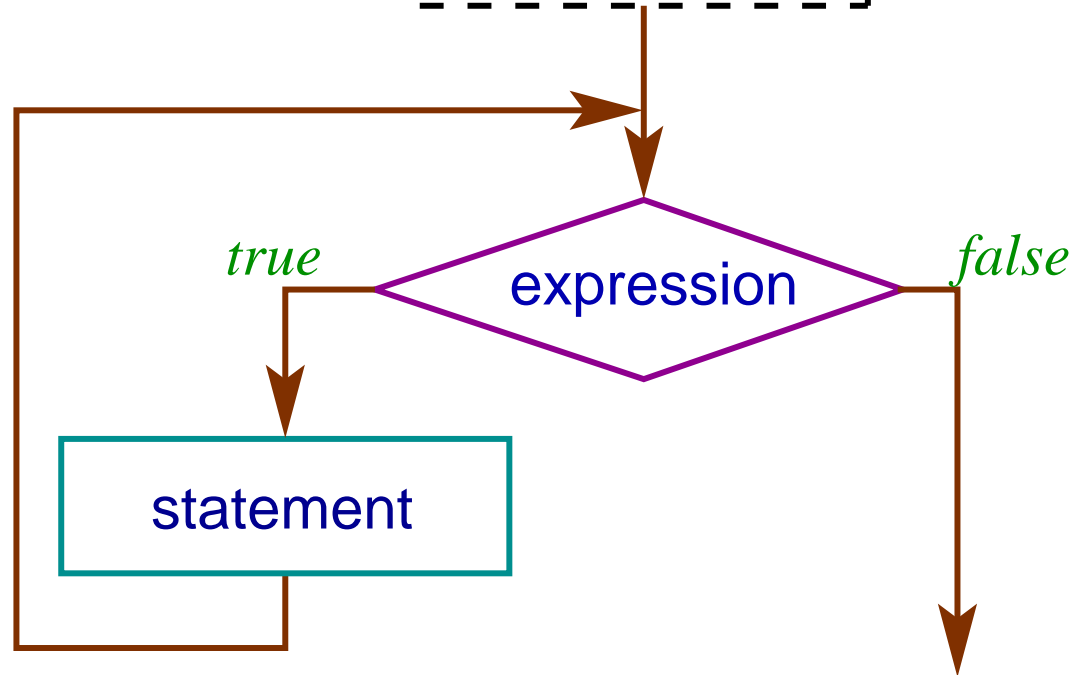
while Statement

while statement of C is one of the constructs used for iterative computation. The **syntax** of the **while** statement is

```
while (expression) statement
```

while loop

Initialization



Example

```
#include <stdio.h>
int main(){ // whileExpl.c
    int n, data, sum;
    printf("Enter a +ve integer: ");
    scanf("%d", &n);
    printf("Enter %d data:\n", n);
    sum = 0; // Initialisation
    while(n > 0){ // Condition
        scanf("%d", &data); // Statements
        sum = sum + data;
        --n;
    }
```

```
printf("Sum: %d\n", sum);  
return 0;  
}
```

Tutorial II.4

Modify the code of `whileExpl.c` to compute the geometric mean (GM) of the data (assume all data to be positive). Use the mathematical library function `pow(x, y)`, that computes x^y . Geometric mean of n data is,

$$GM = \sqrt[n]{d_1 \times \cdots \times d_n}$$

Use `%f` format for input/output of `float` data.

scanf() returns

```
#include <stdio.h>
int main() // scanfRet.c
{
    int a, b, c, val;

    val = scanf("%d%d%d", &a, &b, &c);
    printf("val: %d\n", val);
    return 0;
}
```

scanf() returns the number of data read or EOF.

Tutorial II.5

Modify the code of `tutII.4.c` where the data count is not known a priori. Use the return value of `scanf()`.