MA 511: Computer Programming

Lecture 4

http://www.iitg.ernet.in/psm/indexing_ma511/y08/index.html

Partha Sarathi Mandal

psm@iitg.ernet.ac.in

Dept. of Mathematics, IIT Guwahati

Semester 1, 2008-09

Mon 10:00-10:55 Tue 11:00-11:55 Fri 9:00-9:55 Class: 1G2

MA512 Lab : Wed 14:00-16:55

Type Casting

- The value of an expression can be converted to a different data type if desired follows
- (*data type*) *expression*
- Example:

  int i = 7;

  float f = 8.5;

  (i+f)%4 = invalid since i+f is floating point.

  ((int) (i+f)%4 =3

Variable

- Identifier, its used to represent some specific type (single data item) of information within a portion of program.

```
main(){
        int a, b, c;
        char d;
                        a = 2;
                        b = 1;
                        d = ‘u’
                        c= a+b;
                        …..

                        a = 6;
                        b = 10;
                        d = ‘X’
                        c = a*b;
}
```

Unary operators

- Two unary operators:
  - Increment operator: ++ (increased by 1)

    ++i equivalent i=i+1 //

    i++ //value of the operand will altered after it is utilized

    Example:

    a=10; b=20;

    x= ++a;

    y=b++;

    printf("x = %d a = %d\n", x, a) : x = 11 a = 11

    printf("y = %d b = %d\n", y, b) : y = 20 b = 21

    x = ++a equivalent to the following two sequence a = a + 1; then x = a;

    Y = b++ equivalent to the following two sequence y = b; then b = b + 1;

    X = a++ - ++a; undefined [dependent on compiler's implementation]
  - Decrement operator: -- (decreased by 1)

    --i equivalent i=i-1 and i--
  - sizeof(*type*): Example: printf("integer: %d\n", sizeof(float));

Relational and logical operators

Operator	Meaning
<	less than
<=	less than or equal to
>	grater than
>=	grater than or equal to
==	equal to
!=	not equal to
||	or
&&	and

Conditional Statements

- int i = 7;
- float f = 5.5;
- char c = ‘w’ [w= 119 (ASCII)]
- Expression: (i>=6)&&(c==‘w’)
  (i>=6)||(c==119)
  (f<11)&&(i>100)
  (c!=‘p’)||((i+f)<=10)

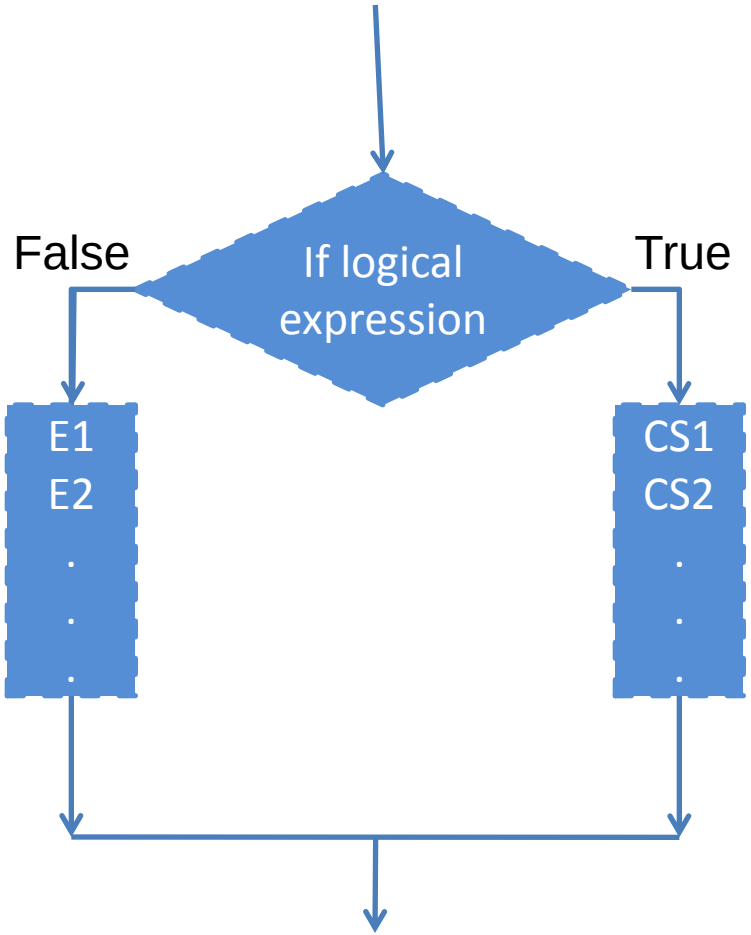
```
if(logical expression){
    CS1
    CS2
      …
} /* end of if */
```

Hierarchy of operator precedence

Operator Category	Operators	Associativity
Unary operators	++ -- ! sizeof (type)	R -> L
Arithmetic multiply, divide and remainder	* / %	L -> R
Arithmetic add and subtract	+ -	L -> R
Relational operators	< <= > >=	L -> R
Equality operators	== !=	L -> R
Logical and	&&	L -> R
Logical or	||	L -> R
Assignment operators	= += -= *= /= %=	R -> L

- i >= 6 && c == 'w'
- i >= 6 || c ==119
- f < 11 && i > 100
- c != 'p' || i+f <= 10

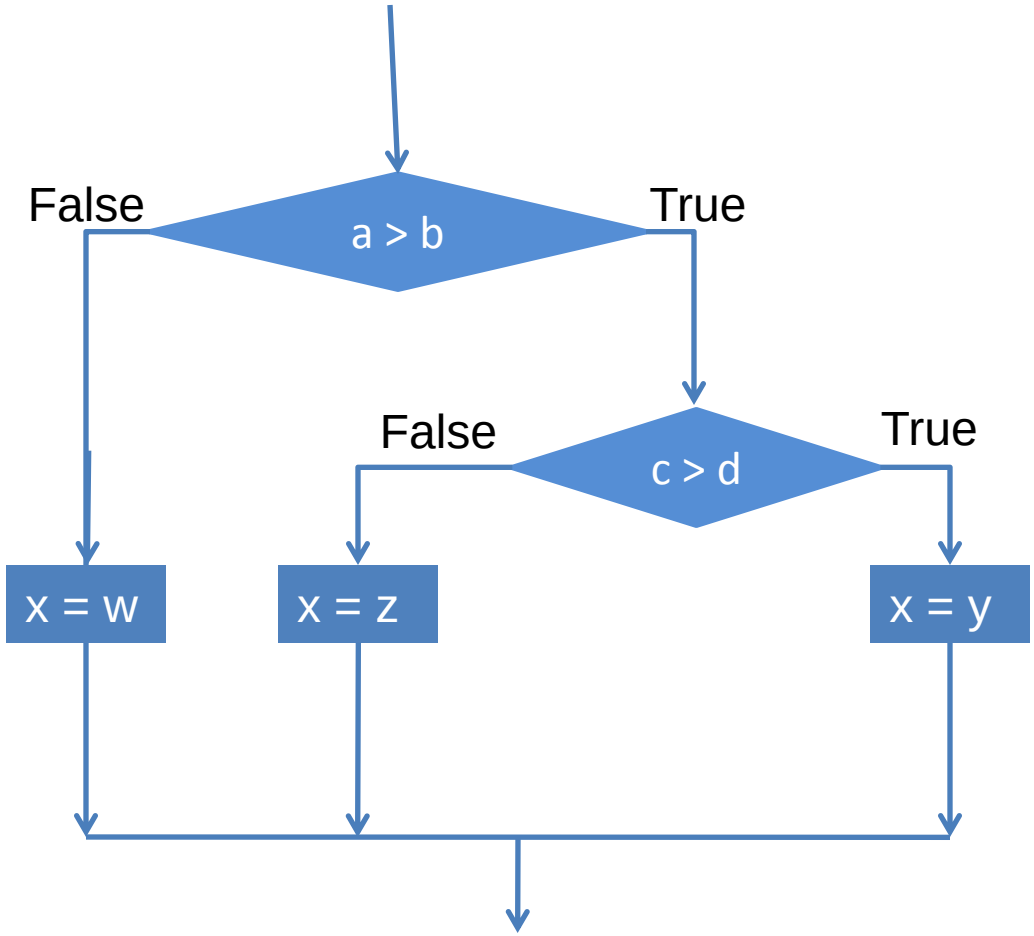
Cont.. Conditional statements




```
if(logical expression){
        CS1
        CS2
          …
} /* end of if */
```

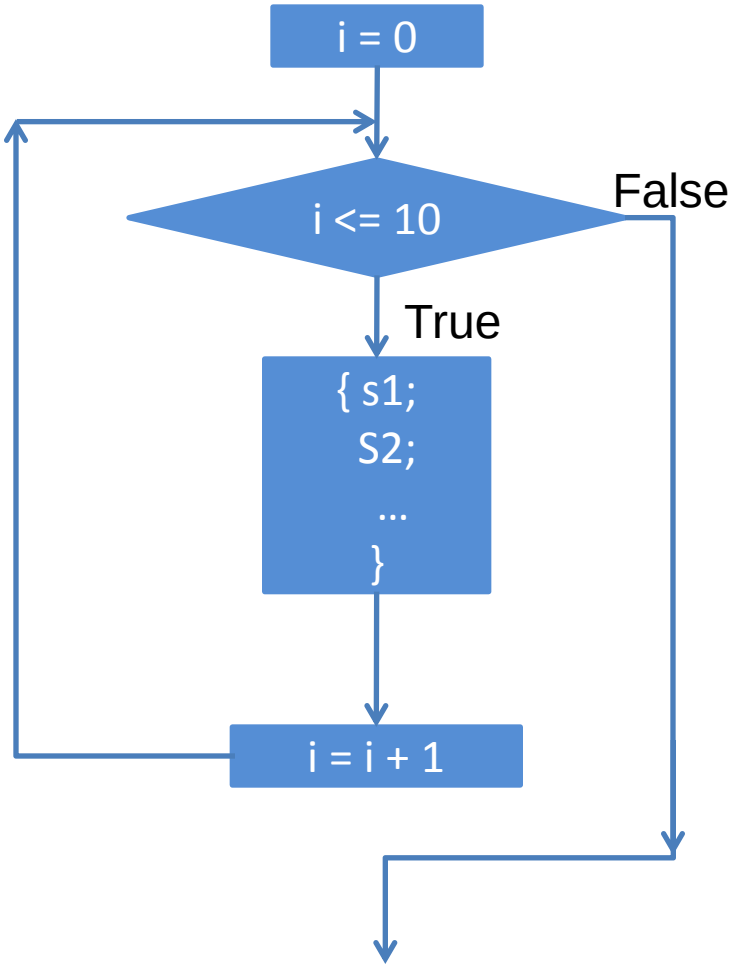
```
if(logical expression){
        CS1
        CS2
          …
}
else{
        E1
        E2
          …

} /* end of if */
```

Nested conditional statement



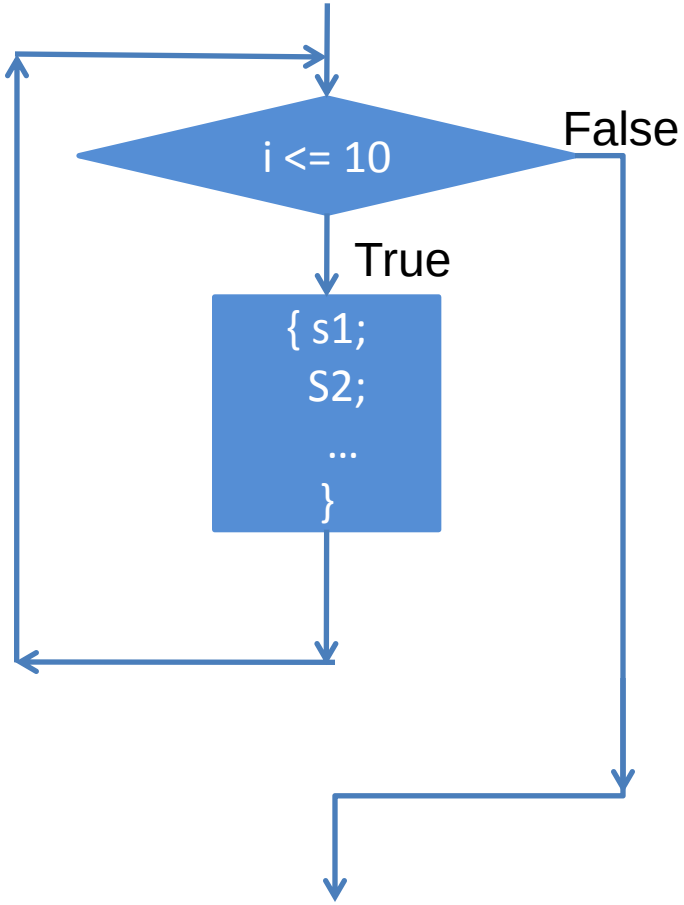

```
if(a > b){
        if(c>d){
            x=y;
        }
        else{
            x=z;
        }
else{
    x=w;
}
} /* end of if */
```

for loop



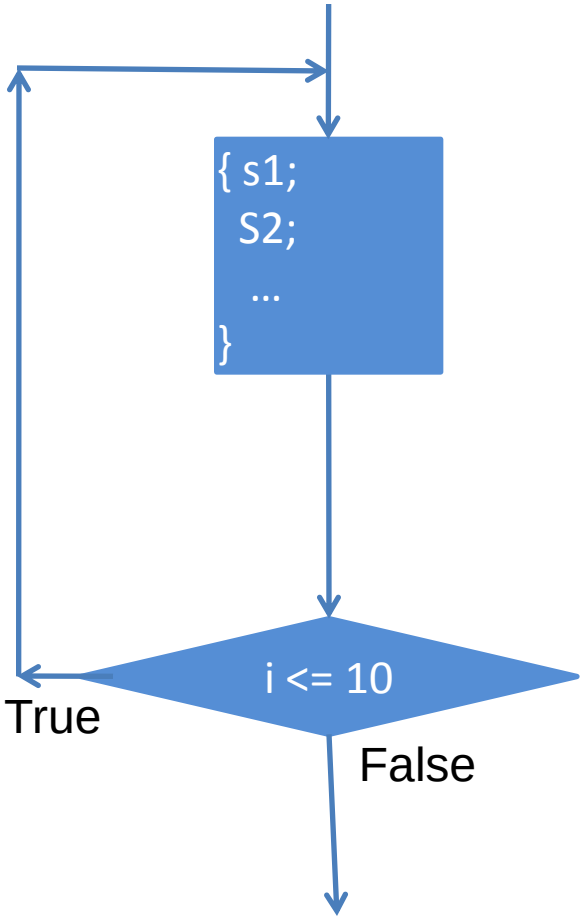

```
for(i = 0; i <= 10; i= i+1){
        s1
        s2
        …

} /* end of for */
```

while loop




```
while(i <= 10){
        s1
        s2
        …

} /* end of while */
```

do while loop




```
do {
        s1
        s2
        …

}
while(i <= 10);
```

Assignment operators

- Five additional assignment operators:

  +=, -=, *=, /=, %=

  expression1 @ = expression2

  is equivalent to

  expression1 = expression1 @ expression2

  Where @ = +, -, *, /, %

  Example: i+=5 is equivalent to i = i + 5

  i%=(j-2) is equivalent to i = i % (j-2)

Conditional operator

- *expression1* ? *expression2* : *expression3*

Example:

1. (i<0) ? 0 : 100
   - expression i<0 is evaluated first. If it is *true* the entire conditional expression takes on the value 0 otherwise 100.
2. min = (f<g)? f : g
3. c += (a>0 && a <= 10) ? ++a : a/b;

Array

- Variable identifier, refers to a collection of the same type of data items that all have the same name.
- individual data items are represented by their corresponding array elements.
- Example:

```
int x[5]; // x[5] is basically x[0], x[1], …, x[4]
float A[10], B[4][5];
for(i=0; i< 10; i=i+1){
    scanf("%d", &A[i]);
}
for(i = 1; i <= 4; i = i + 1 ){
    for(j = 1; j <= 5; j = j + 1){
       scanf("%d", &B[i][j]);
    }
}
```

Assignments

- Write c-codes for
  1. Write c-code for testing two given integer are relatively prime.
  2. Write a program for testing a given integer is a Fibonacci number.
  3. Sin$x = x - x^3/3! + x^5/5! - x^7/7! + \ldots$ x is in radians
     i. c-code for sum of the first n terms (input x and n)
     ii. Adding successive terms in the series until the value of the next term smaller than 10^{-5} in magnitude.
  4. Addition of two mxn matrices.
  5. Multiplication of mxk and kxn matrices.
  6. Transpose a square matrix.
  7. Multiplication of two polynomial of degree m and n respectively where coefficients are integer.