Expressions
&
Control Structures

Objectives of this session

- Expressions and Their Types
- Special Assignment Expressions
- Control Structures
  - If statement
  - Switch statement
  - Do-while, While and For statement

Expressions and Their Types

- **Constant Expressions**
- **Integral Expressions**
- **Float Expressions**
- **Pointer Expressions**
- **Relational Expressions**
- **Logical Expressions**
- **Bitwise Expressions**

An expression may also use combination of the above expressions – ***Compound expressions.***

Constant Expressions

Constant Expressions consist of only constant value.

Eg:-

15

20 + 5 / 2.0

‘ X ’

Integral Expressions

Integral Expressions are those which produce integer results after implementing all the automatic and explicit type conversions.

Eg:-

m

m * n – 5

m * ‘ x ’

5 + int(2.0)

where m and n are integer variables.

Float Expressions

Float Expressions are those which, after all conversions, produce floating-point results.

Eg:-

x + y

x * y / 10

5 + float(10)

10.75

where x and y are floating-point variables.

Pointer Expressions

Pointer Expressions produce address values.

Eg:-

&m

ptr

ptr + 1

“xyz”

where m is a variable and ptr is a pointer.

Relational Expressions

Relational Expressions yield results of type bool which takes a value true or false.

Also known as ***boolean expressions***.

Eg:-

$x <= y$

$a + b == c + d$

$m + n > 100$

When arithmetic expressions are used on either side of a relational operator, they will be evaluated first and then the results compared.

Logical Expressions

Logical Expressions combine two or more relational expressions and produces **bool** type results.

Eg:-

a > b && x == 10

x == 10 | | y == 5

Bitwise Expressions

Bitwise Expressions are used to manipulate data at bit level. They are basically used for testing or shifting bits.

Eg:-

x << 3 // Shift three bit positions to left

y >> 1 // Shift one bit position to right

Special Assignment Expressions

Chained Assignment

```
x = (y = 10);   // first 10 is assigned to y
```

or

```
x = y = 10;    // and then to x
```

A chained statement can not be used to initialize variables at the time of declaration.

```
float a = b = 12.34              // wrong
float a = 12.34, b = 12.34   //  correct
```

Special Assignment Expressions

continue…

Embedded Assignment

x = (y = 50) + 10;

Here the value 50 is assigned to y and then the result 50 + 10 = 60 is assigned to x.

This statement is identical to

y = 50;

x = y + 10;

Special Assignment Expressions

continue…

Compound Assignment

A combination of the assignment operator with a binary operator.

x + = 10;   += is known as compound operator

variable_1 op= variable_2

where op is a binary arithmetic operator

Control Structures

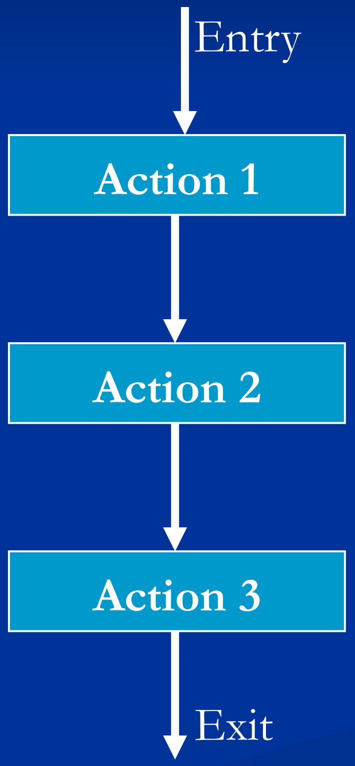
- □ ***Sequence Structure (straight line)***
- □ ***Selection Structure (branching)***
- □ ***Loop Structure (iteration or repetition)***

Structured programming – The approach of using one or more of these basic control constructs in programming.

Control Structures

continue…

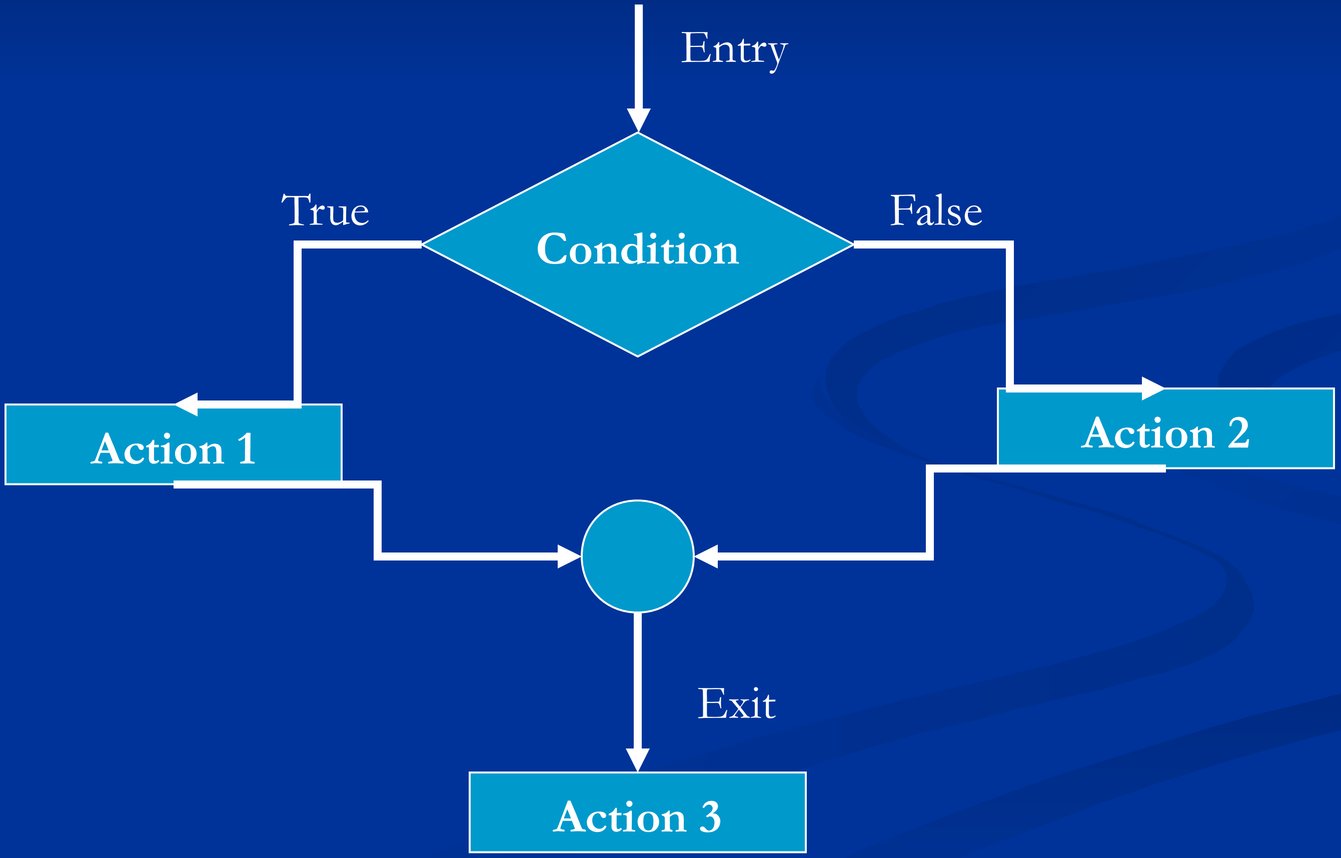
- ***Sequence Structure (straight line)***



Control Structures

continue…

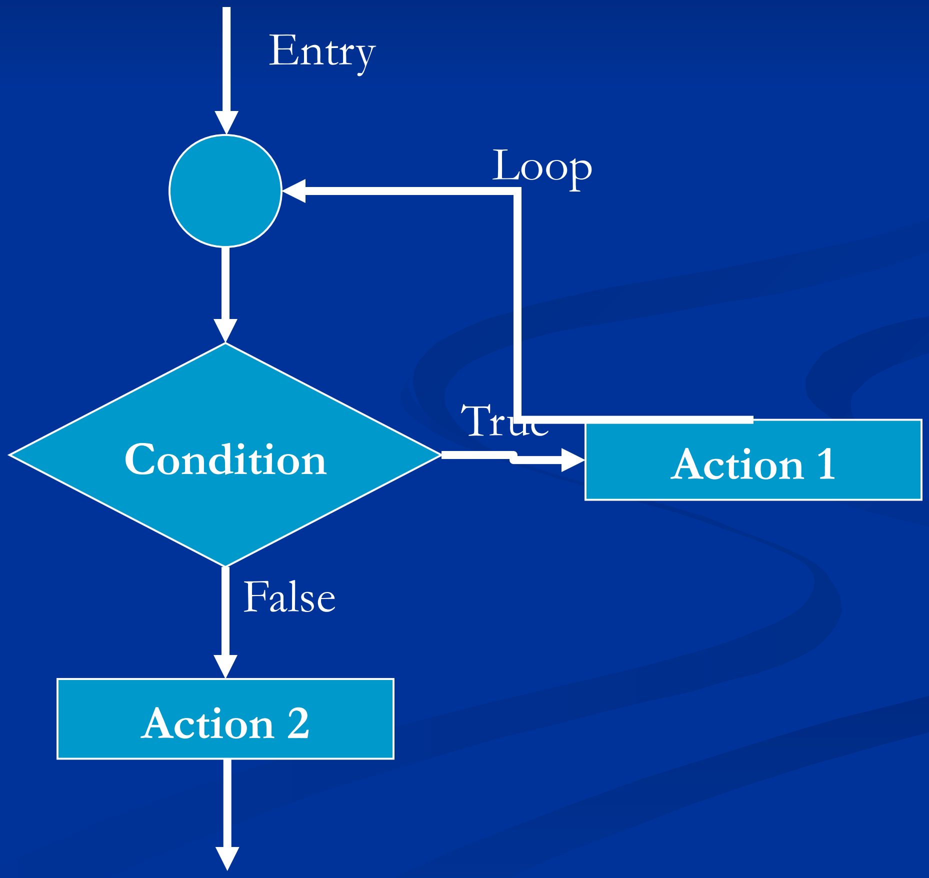
- □ ***Selection Structure (branching)***



Control Structures

continue…

- □ ***Loop Structure (iteration or repetition)***



If Statement

The if statement is implemented in two forms:

- Simple if statement

```
if (expression is true)
{
    action 1;
}
action 2;
```

If Statement

continue…

- □ if … else statement

```
if (expression is true)
{
     action 1;
}
else
{
     action 2;
}
action 3;
```

The Switch Statement

```
switch (expression)
{
        case 1:
        {
                action 1;
        }
        case 2:
        {
                action 2;
        }
        default:
        {
                actiion 3;
        }
}
action 4;
```

Thank You