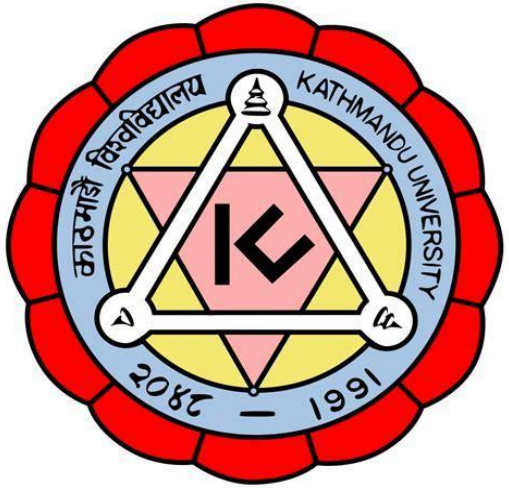
KATHMANDU UNIVERSITY SCHOOL OF MANAGEMENT

BBIS

COM 102 : 3 Credit Hours



6. Control structures and statements in C

Outlines

- 6.1 Conditional Statements
  - a. if Statement
  - b. if-else Statement
  - c. switch statement
- 6.2 Loop Statements
  - a. for loop
  - b. while loop
  - c. do-while loop
- 6.3 Break Control Statements
  - a. break
  - b. continue
  - c. go-to statement

Intro…

Control structures are used to alter the flow of execution of the program.

Why do we have to alter the program flow ?

For -“decision making“!

With the choice we make in our life, we alter the flow of our life’s direction.

Three sorts of control structures in C

1) Sequence structures (straight line paths)

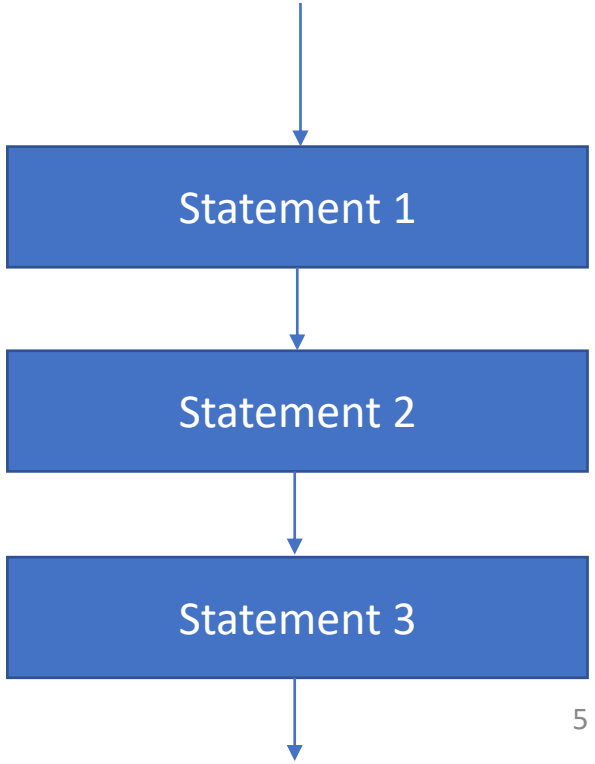
2) Selection structures (one or many branches)

3)Loop structures (repetition of a group of activities)

Sequence structures

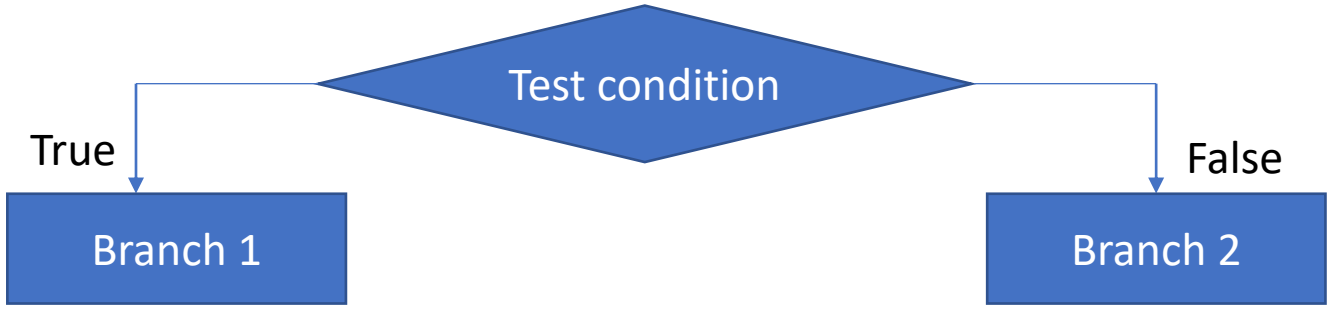
- the statements of a program are executed in the same order in which they are written in the program.
- The statements are executed one after the other.
- All statements of the program executed sequentially without skipping any statement.

The first statement is “Statement-1” and “Statement-n” is the last statement that will be executed in a sequence from top to bottom (such as from statement-1 to statement-n).



Selection structures

- Program should be able to make logical decisions (True/False) based on given condition
  - And for this we use decision making statements for this purpose
- Selection structures are used to perform ‘decision making‘ and then branch the program flow.
- Selection structures are implemented in C/C++ with
  - If, If Else and Switch statements.



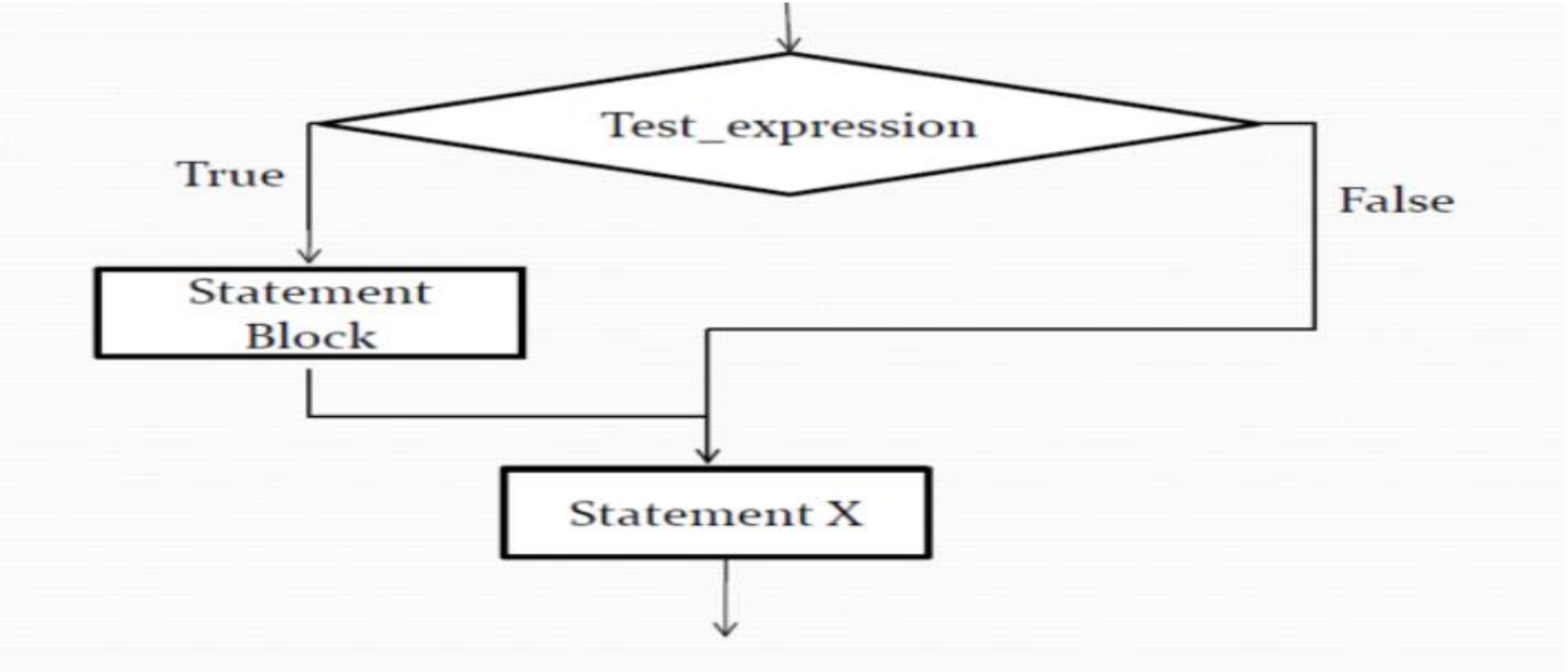
If statement

Syntax:

```
if (a>5)
{
        statement A;
        statement B;
        ....
}
Statement X;
```

- If test_expression is true statement block will be executed, otherwise execution will jump to statement X.

Flowchart of If statement



Example: If statement

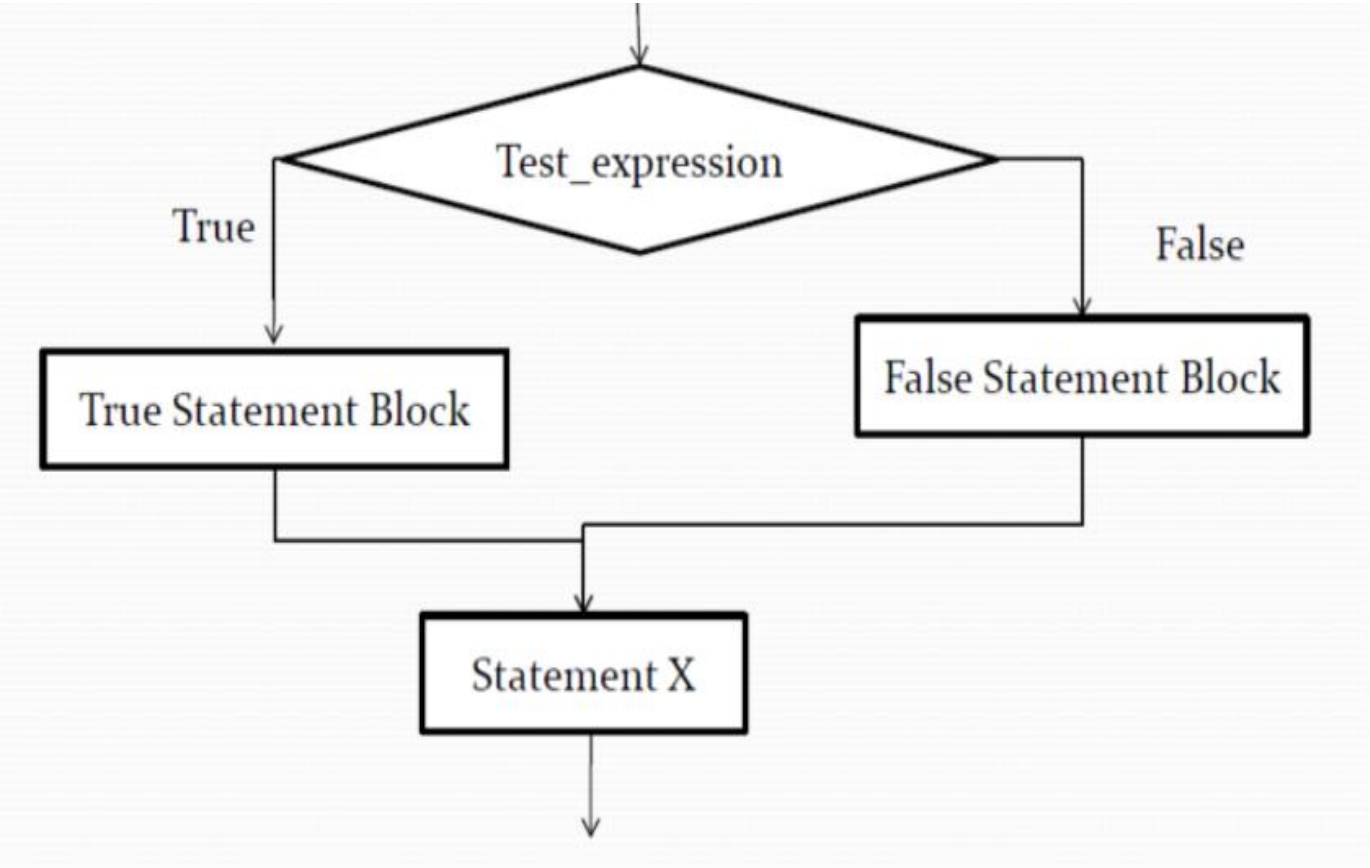
```
If(a>10)
{
        printf(“a is greater than 10”);
}
Printf(“a=%d”,a);
```

If else statement

- Extension of if statement.
- Syntax

```
If(test_expression)
{
        statement A;
        statement B;
        …
}
else
{
        statement M;
        statement N;
        …
}
Statement X;
```

Flowchart



Example: If Else statement

```
If(a>10)
{
        printf(“a is greater than 10”);
}
Else
{
        printf(“a is less than or equal to 10”)
}
Print(“a=%d”, a);
```

Nested if else statement

```
If(16>10)
{
        if(14>15)
                printf(“a is greater than 15”);
        else
                 printf(“a is greater than 10 and and less than 15”);
}
```

Outer if-statement block

```
Else
{
        if(a==10)
                printf(“a is  equal to 10”);
        else
                 printf(“a is  less to 10”);
}
```

Outer else-statement block

```
Print(“a=%d”, a);
```

Else-if ladder

- It is another way of putting if's together when multipath decisions are involved.

```
If(2==5)
{
        printf("a is equal to 5");
}
Elseif(a>5)
{
Printf("a is greater than 5");
}
Elseif
{
        printf("a is less than 5");
}
Printf("a=%d",a);
```

```
If(a==5)
        printf("a is equal to 5");
else
{
    if(a>5)
    Printf("a is greater than 5");
    else
        printf("a is less than 5");
    }
Printf("a=%d",a);
```

Classwork

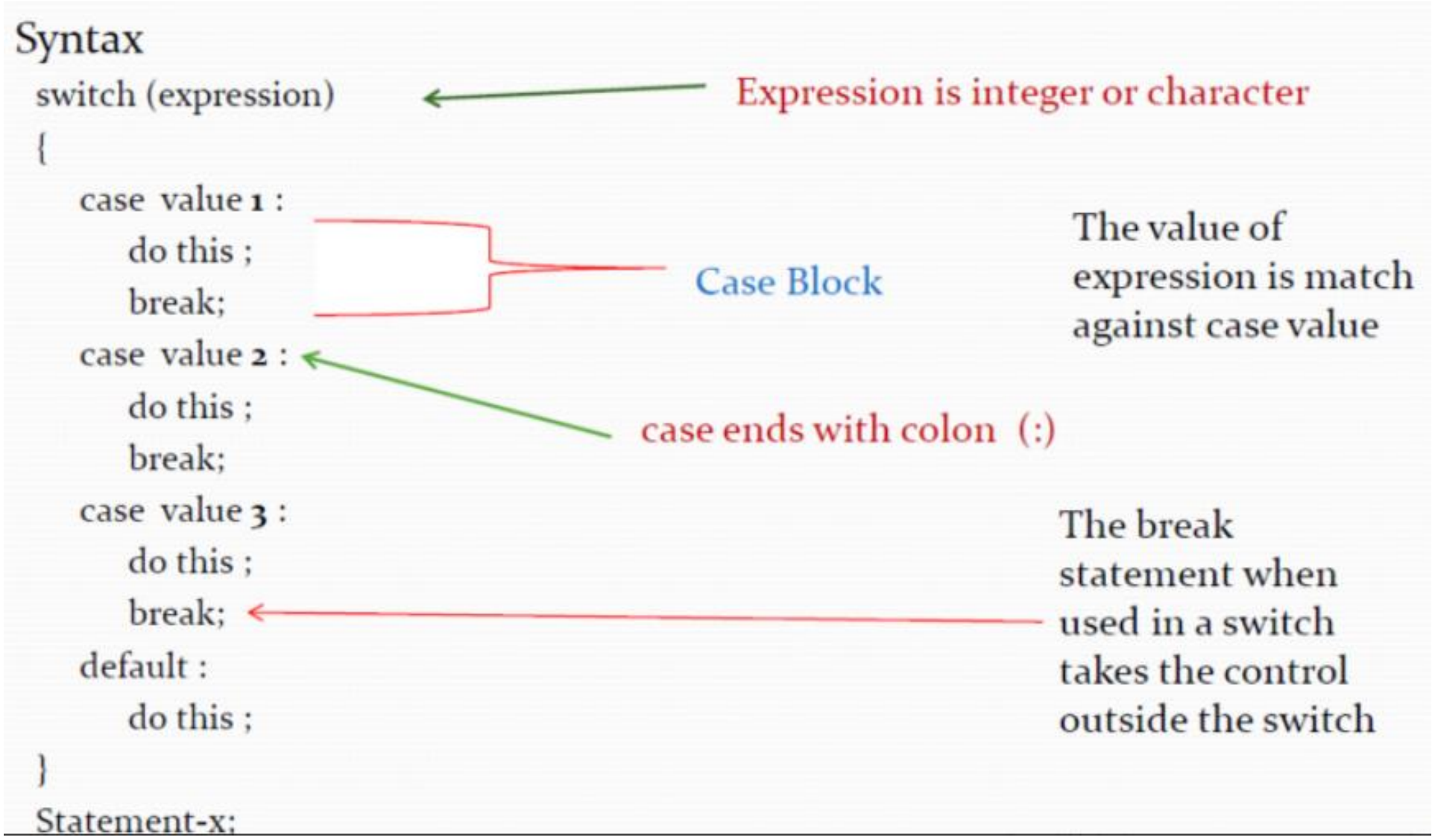
- WAP to read percentage of a student and print the equivalent grade.
- Using switch

  - Between 100 to 90 : grade A
  - Between 89 to 80 : grade A-
  - Between 79 to 70 : grade B-
  - Between 69 to 60 : grade B-
  - Between 59 to 50 : grade c
  - Below 50 : Fail

Switch Statement

- The behavior of the switch case statement is very similar to the ‘if..else if..else’ ladder in C.
- But, the implementation of this same use case/problem using switch statement is much more neat and elegant compared to the ‘if..else if..else’ ladder.

Syntax:


Syntax
switch (expression)
{
case value 1 :
do this ;
break;
case value 2 :
do this ;
break;
case value 3 :
do this ;
break;
default :
do this ;
}
Statement-x;
Expression is integer or character
Case Block
The value of expression is match against case value
case ends with colon (:)
The break statement when used in a switch takes the control outside the switch

Switch statement Example

```
Char single;
Printf(“”Enter a character);
Scanf(“%c”, &single);
Switch(a)
{
        case ‘x’:
                printf(“x”);
                break;
        case ‘y’;
                printf(“y”);
                break;
        case ‘z’;
                printf(“z”);
                break;
        default:
                printf(“Not x and y and z entered \n”);

}
Printf(“done”);
```

- Above example is equivalent to following if-else-if statement

```
char single;
printf("Enter a character");
scanf("%c",& single);

if(single==‘x’)
        printf(“x");
else if(single==‘y’)
        printf(“y");
else if(single==‘z’)
        printf(“z");
else
        printf(“Not x and y and z \n");

printf(“done”);
```

Contd…

- If we do not have a default case,
  - Program falls through the entire switch and continues with the next instruction (if any) that follows the closing brace of switch.
  - Float expression cannot be tested using a switch.

Any queries???

Up next: Loop structures