Programming in C

Dr. Rupali P. Patil

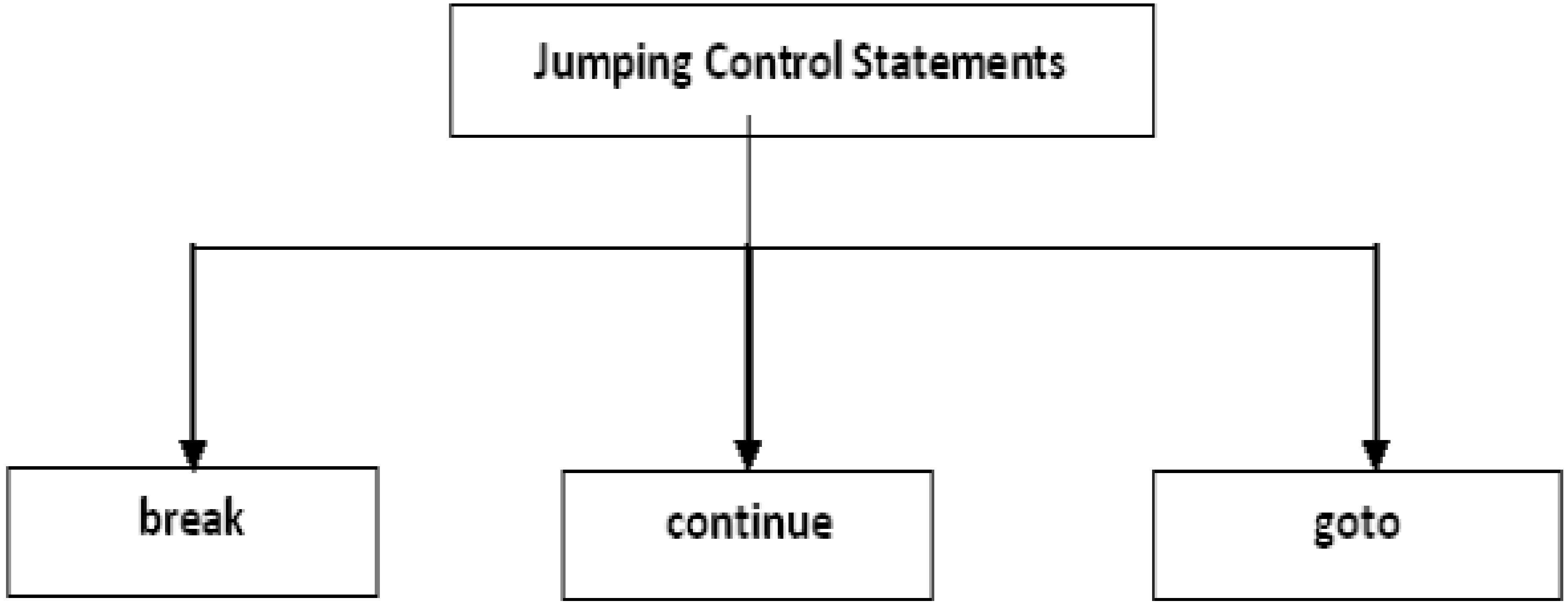
Department of Electronics and Telecommunications

Topics for today

Module 3:
Control Structures in C

Jumping control-flow statements.

Jumping control-flow statements are the control-flow statements that transfer the control to the specified location or out of the loop or to the beginning of the loop. There are 3 jumping control statements:



Jumping control-flow statements.

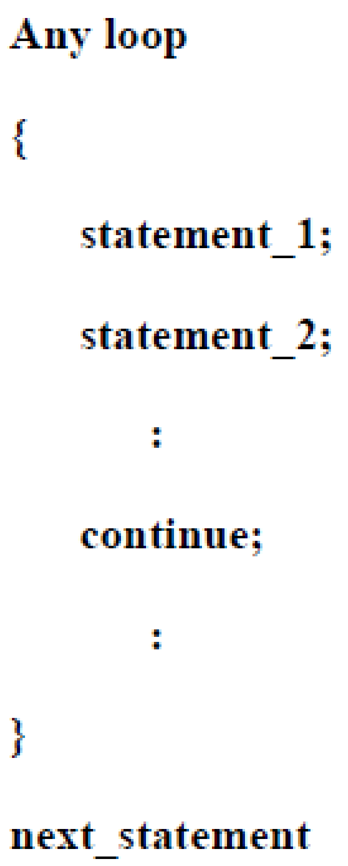
The “break” statement is used with in the looping control statements, switch statement and nested loops. When it is used with the for, while or do-while statements, the control comes out of the corresponding loop and continues with the next statement.

```
Any loop
{
    statement_1;
    statement_2;
        :
    break;
        :
}
next_statement
```

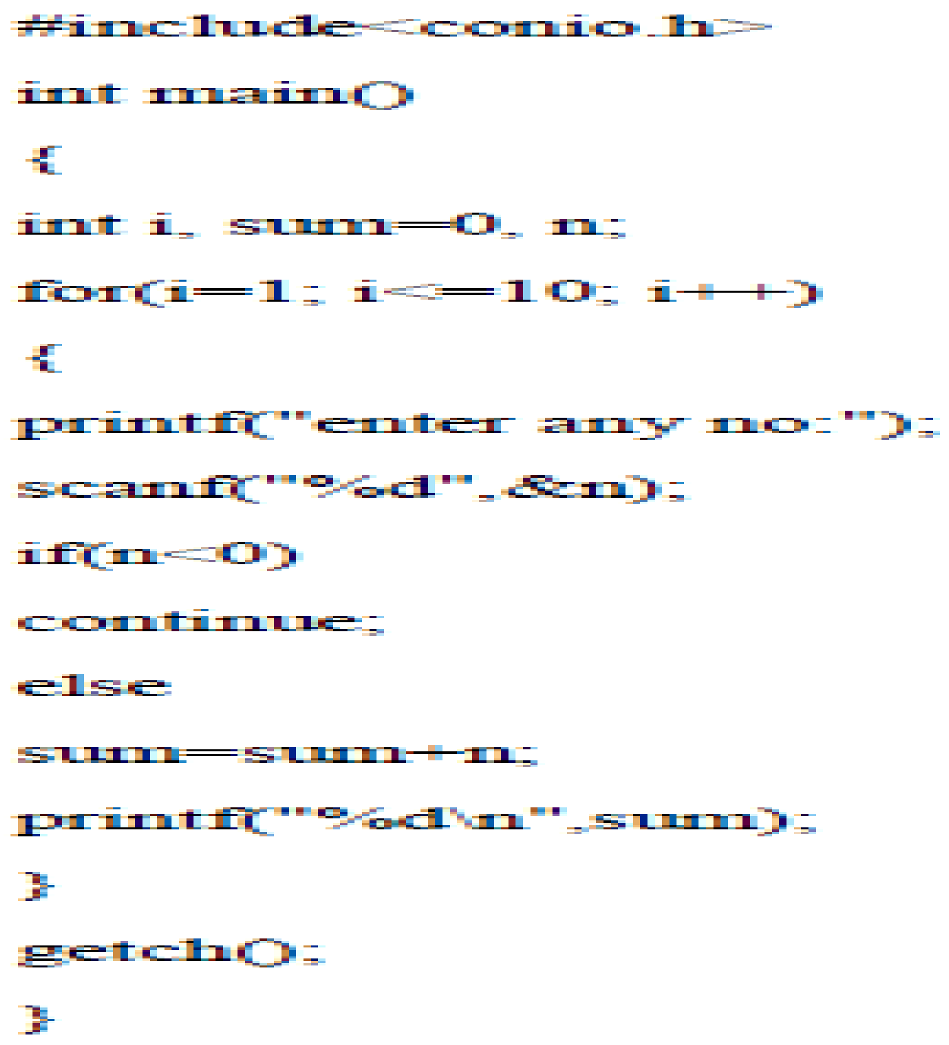
```
#include<stdio.h>
#include<conio.h>
int main()
 {
int i;
for(i=1 ; i<=10; i++)
 {
if(i==6)
break;
printf("%d",i);
 }
 getch();
 }
```

Jumping control-flow statements.

A continue statement is used within loops to end the execution of the current iteration and proceed to the next iteration. It provides a way of skipping the remaining statements in that iteration after the continue statement.

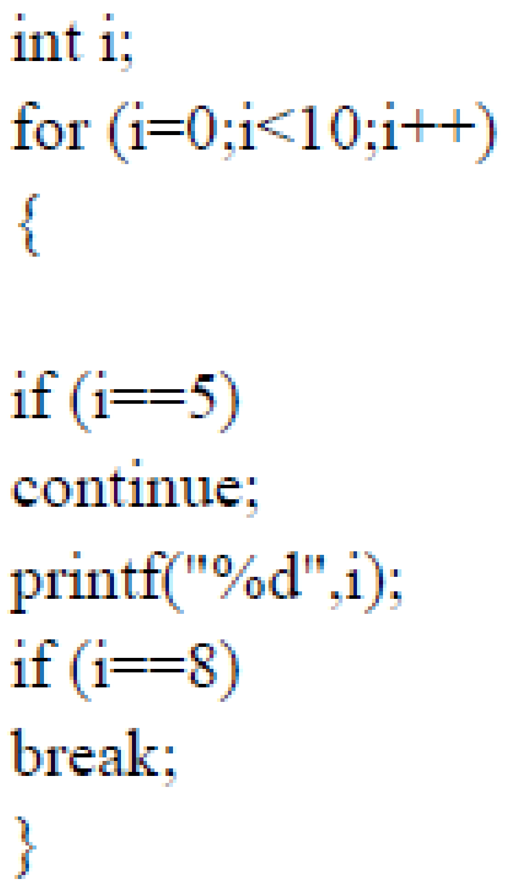


```
Any loop
{
    statement_1;
    statement_2;
        :
    continue;
        :
}
next_statement
```



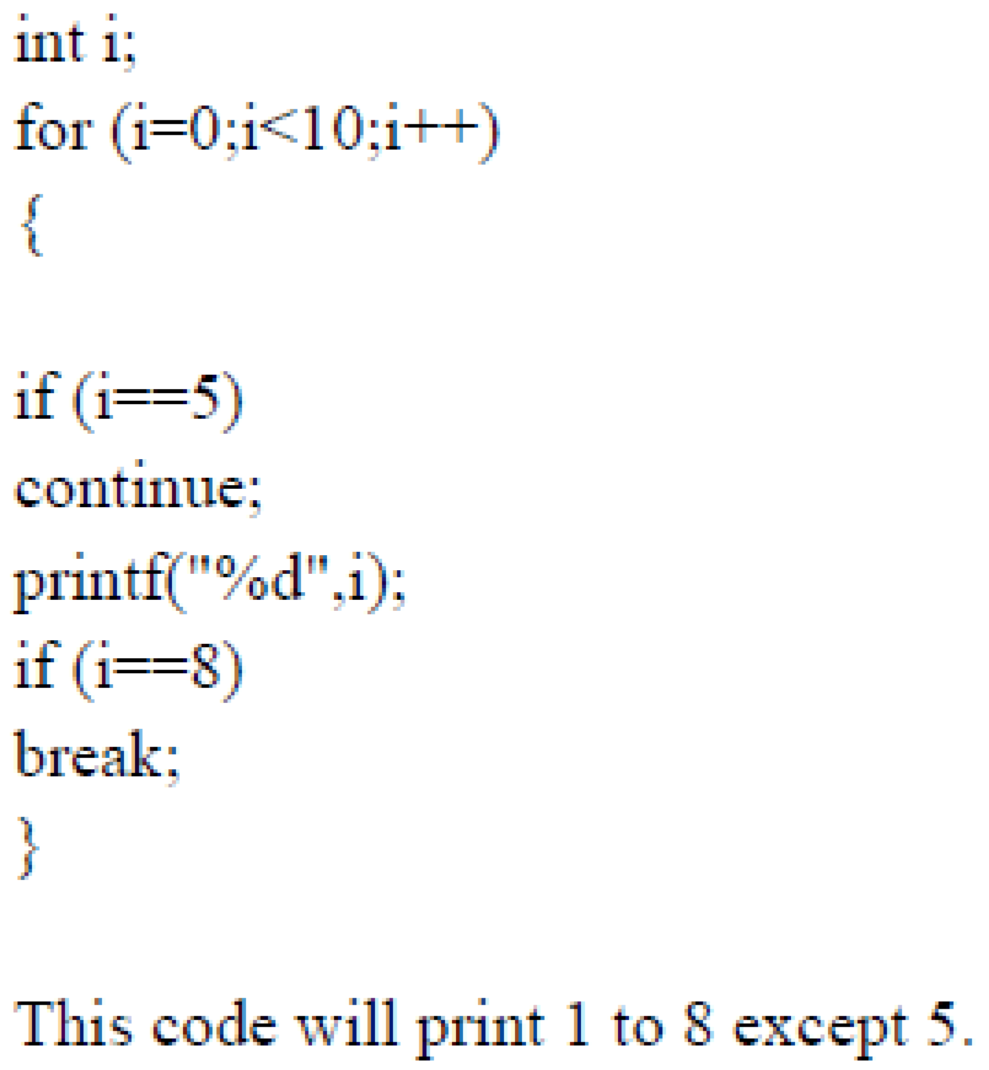
```
#include<conio.h>
int main()
{
int i, sum=0, n;
for(i=1; i<=10; i++)
{
printf("enter any no:");
scanf("%d",&n);
if(n<0)
continue;
else
sum=sum+n;
printf("%d\n",sum);
}
getch();
}
```

Jumping control-flow statements.



```
int i;
for (i=0;i<10;i++)
{

if (i==5)
continue;
printf("%d",i);
if (i==8)
break;
}
```

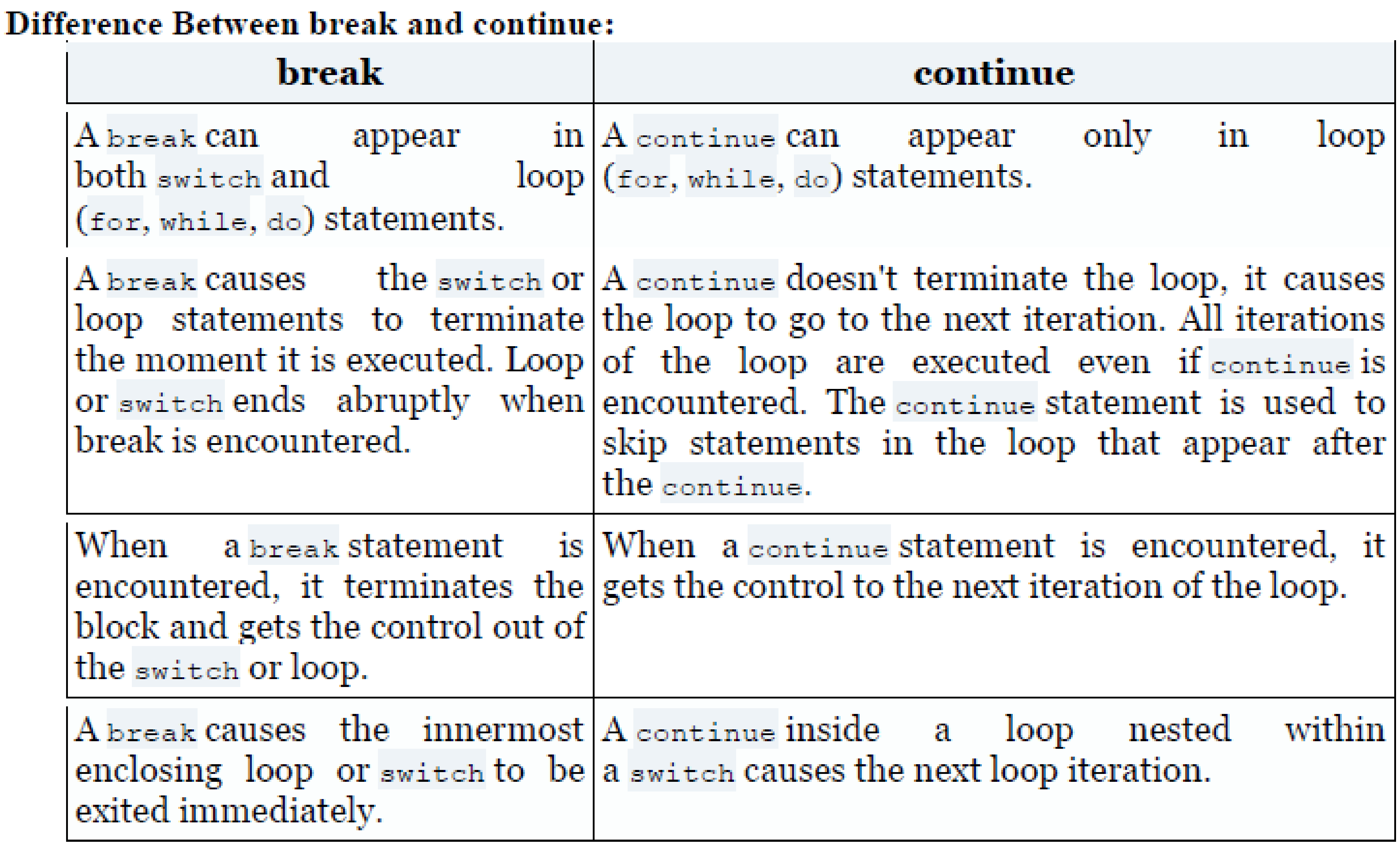
Jumping control-flow statements.



```
int i;
for (i=0;i<10;i++)
{

if (i==5)
continue;
printf("%d",i);
if (i==8)
break;
}
```

This code will print 1 to 8 except 5.

Jumping control-flow statements.



Difference Between break and continue:

break	continue
A `break` can appear in both `switch` and loop (`for`, `while`, `do`) statements.	A `continue` can appear only in loop (`for`, `while`, `do`) statements.
A `break` causes the `switch` or loop statements to terminate the moment it is executed. Loop or `switch` ends abruptly when break is encountered.	A `continue` doesn't terminate the loop, it causes the loop to go to the next iteration. All iterations of the loop are executed even if `continue` is encountered. The `continue` statement is used to skip statements in the loop that appear after the `continue`.
When a `break` statement is encountered, it terminates the block and gets the control out of the `switch` or loop.	When a `continue` statement is encountered, it gets the control to the next iteration of the loop.
A `break` causes the innermost enclosing loop or `switch` to be exited immediately.	A `continue` inside a loop nested within a `switch` causes the next loop iteration.

Jumping control-flow statements.

The goto statement transfers the control to the specified location unconditionally. There are certain situations where goto statement makes the program simpler. For example, if a deeply nested loop is to be exited earlier, goto may be used for breaking more than one loop at a time. In this case, a break statement will not serve the purpose because it only exits a single loop.

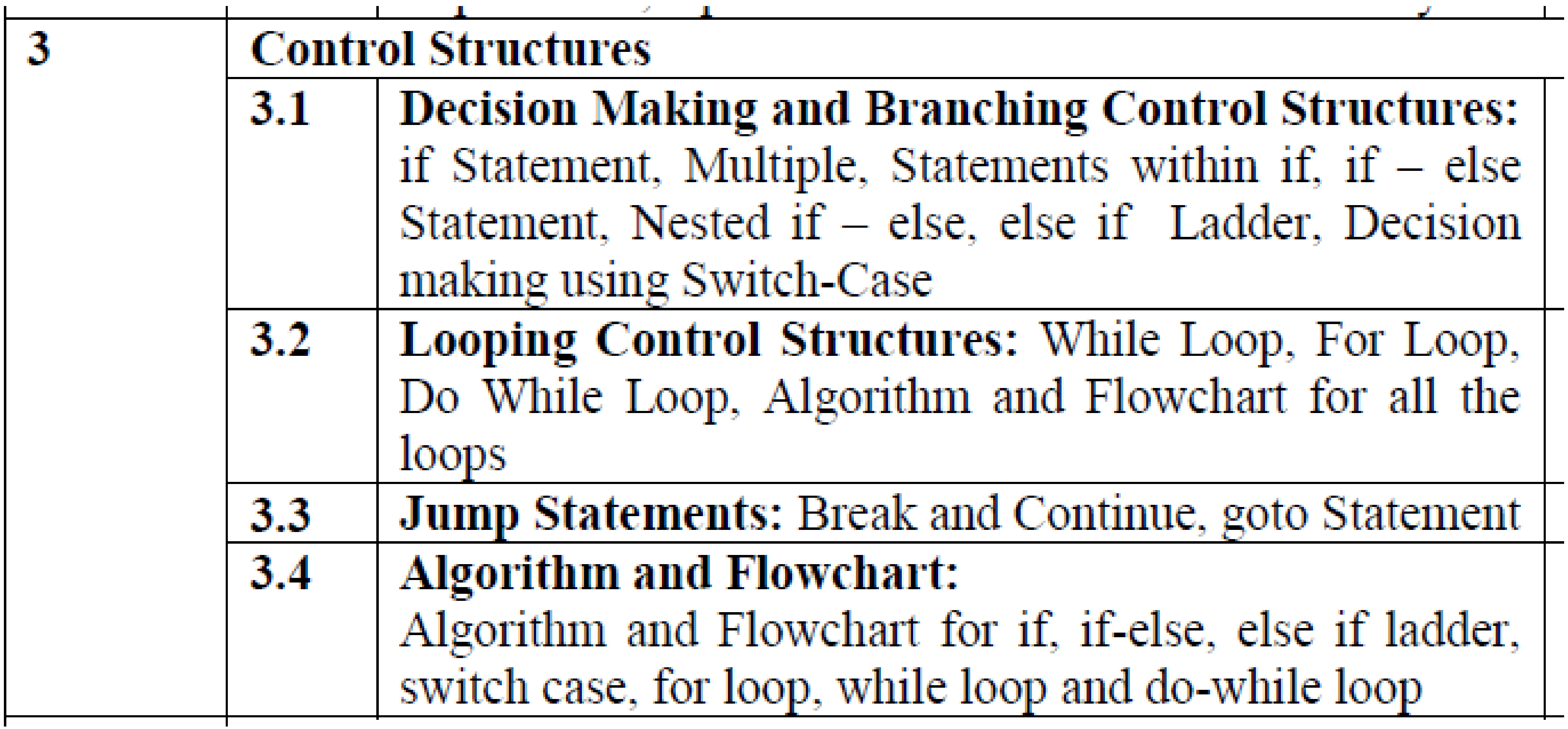
```
label:
        {
        statement_1;
        statement_2;
                :
        }
        :
    goto label;
```

- In this syntax, goto is the keyword and label is any valid identifier and should be ended with a colon (:).
- The identifier following goto is a statement label and need not be declared. The name of the statement or label can also be used as a variable name in the same program if it is declared appropriately.

Jumping control-flow statements.

PROGRAM FOR GOTO STATEMENT:

```
#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

printf("www.");

goto x;

y:

printf("expert");

goto z;

x:

printf("c programming");

goto y;

z:

printf(".com");

getch();

}
```



3	Control Structures	
	3.1	**Decision Making and Branching Control Structures:** if Statement, Multiple, Statements within if, if – else Statement, Nested if – else, else if Ladder, Decision making using Switch-Case
	3.2	**Looping Control Structures:** While Loop, For Loop, Do While Loop, Algorithm and Flowchart for all the loops
	3.3	**Jump Statements:** Break and Continue, goto Statement
	3.4	**Algorithm and Flowchart:** Algorithm and Flowchart for if, if-else, else if ladder, switch case, for loop, while loop and do-while loop