# CSE102
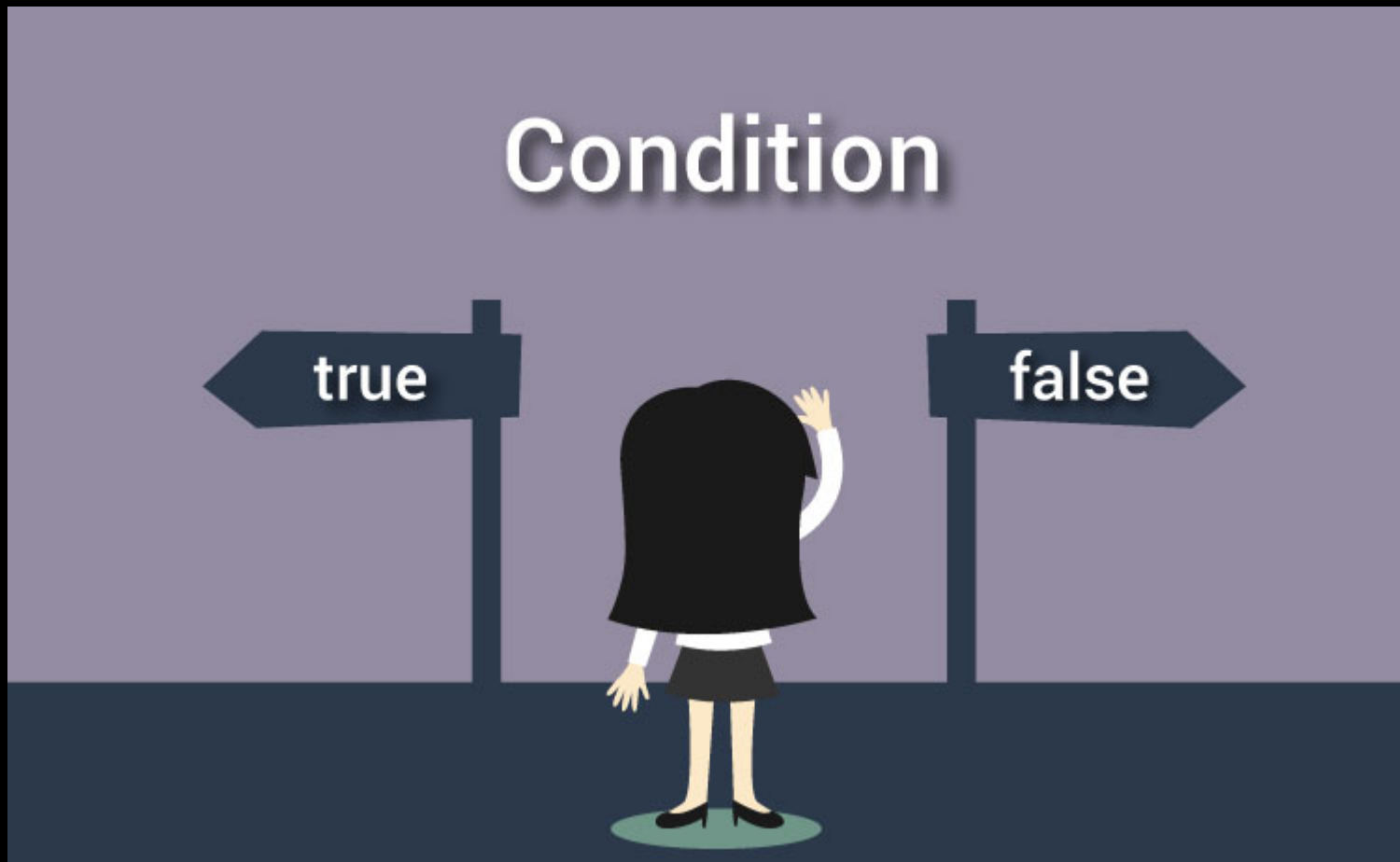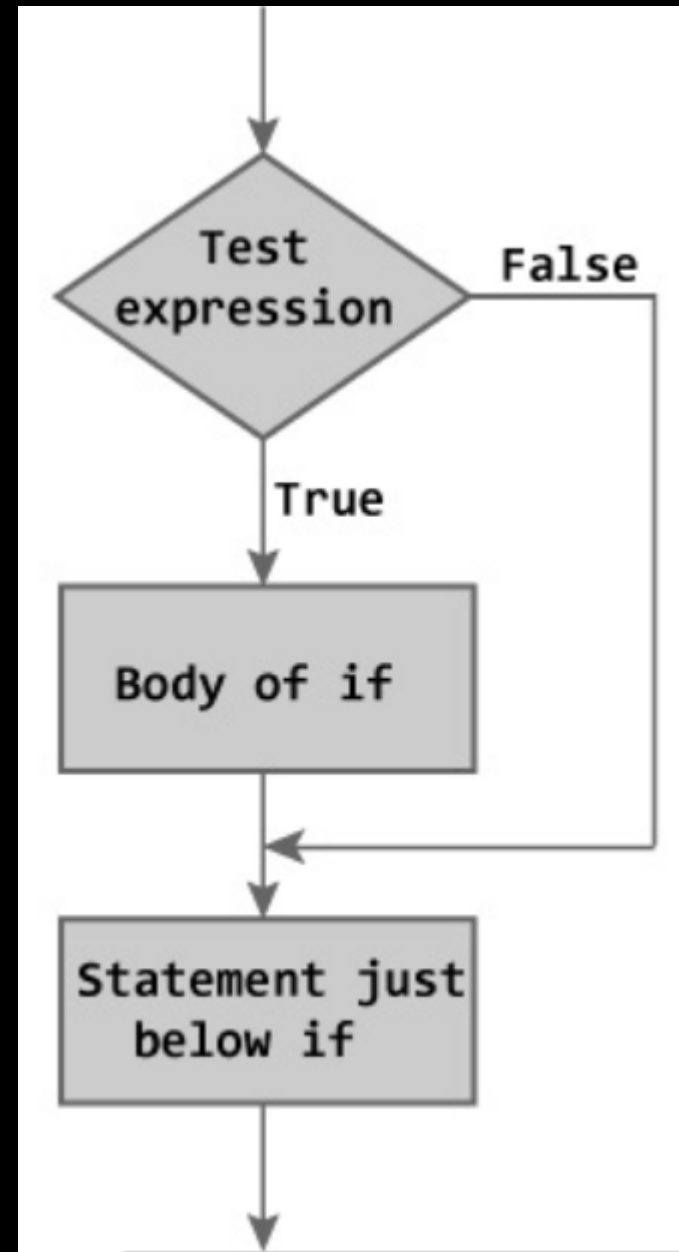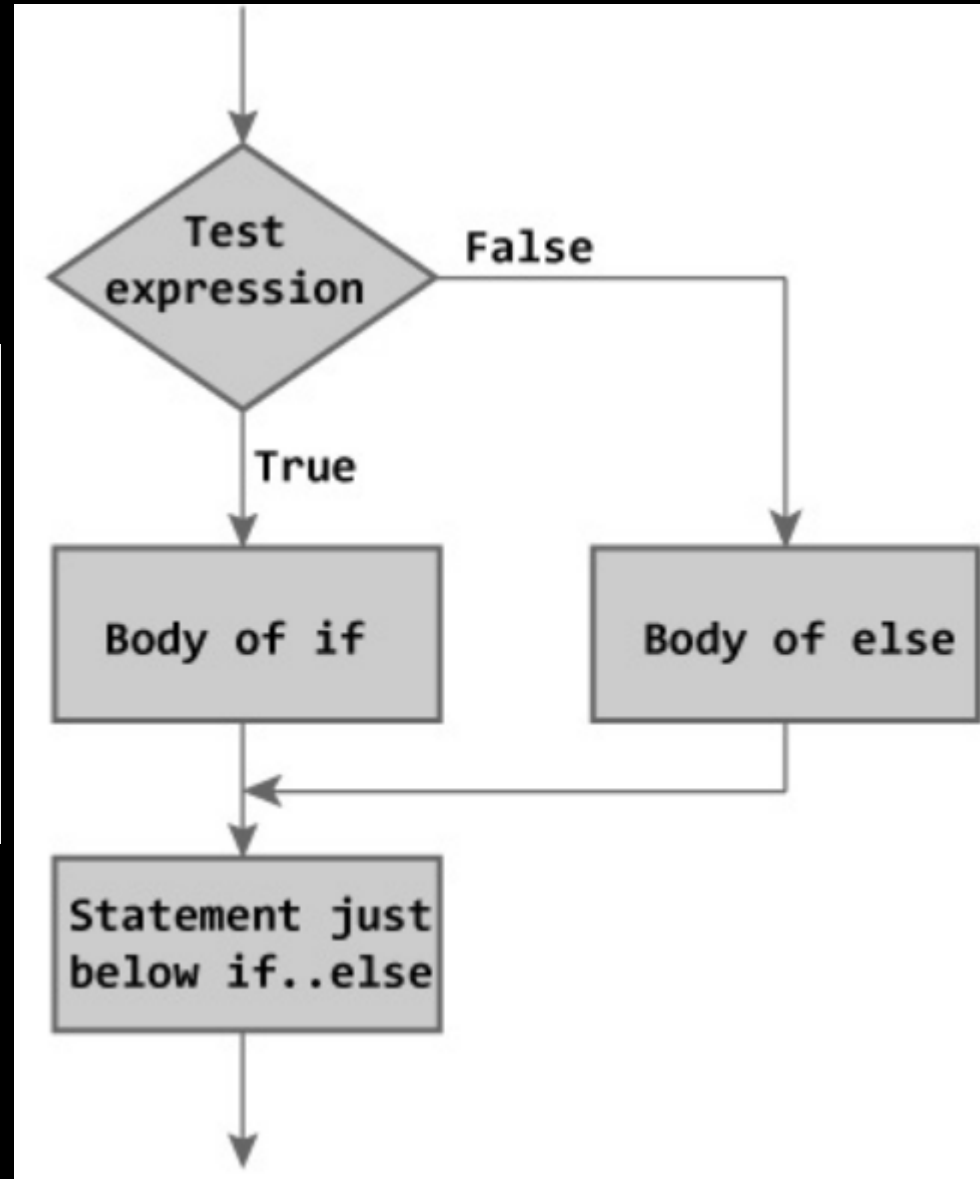# Computer Programming

# Predictable yet Unknown

```
if (testExpression)
{
    // statements
}
```

# Predictable yet Unknown

```
if (testExpression) {
    // codes inside the body of if
}
else {
    // codes inside the body of else
}
```

# Predictable yet Unknown

```
switch (n)
{
    case constant1:
        // code to be executed if n is equal to constant1;
        break;

    case constant2:
        // code to be executed if n is equal to constant2;
        break;

        .

        .

        .

    default:
        // code to be executed if n doesn't match any constant
}
```
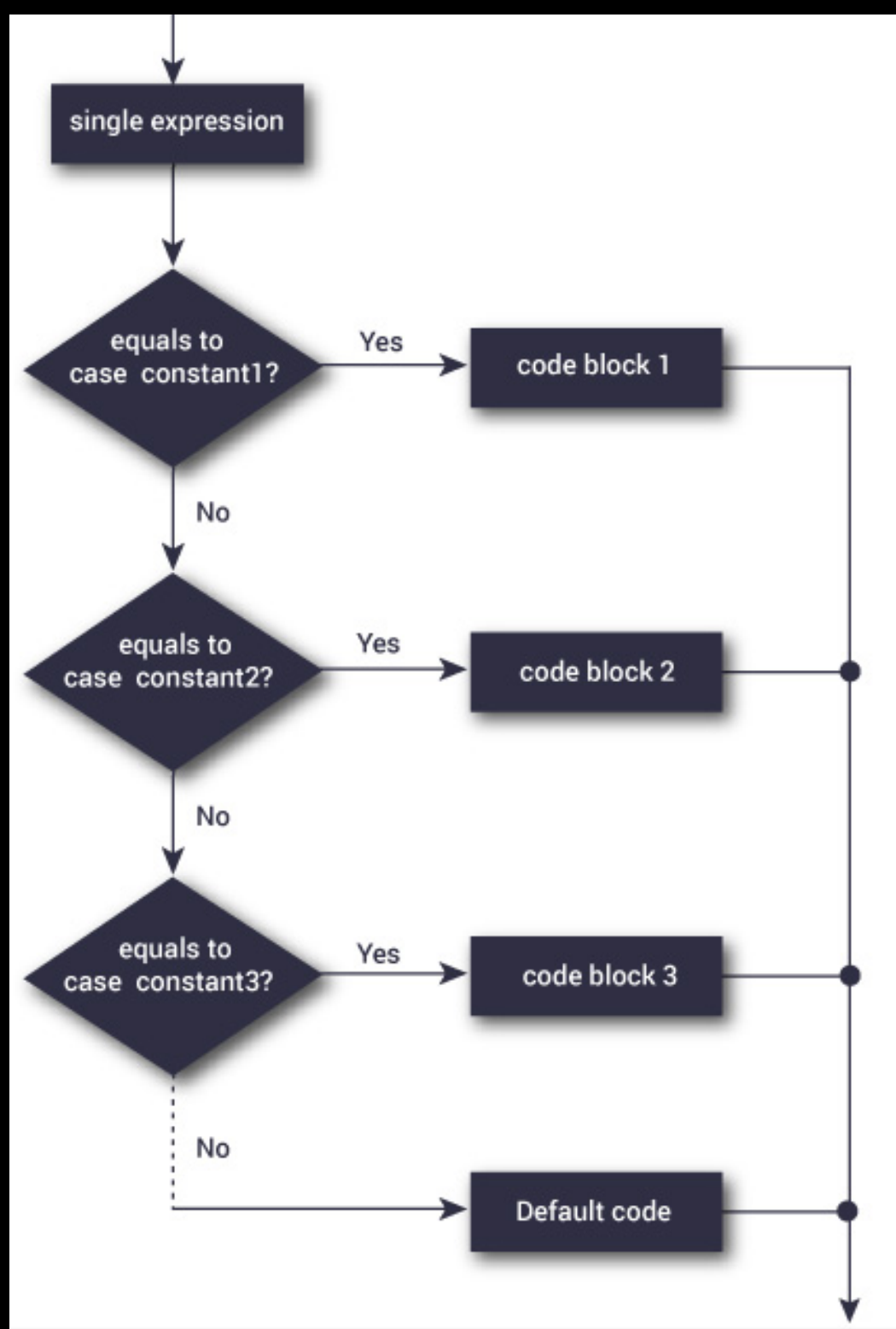
# Predictable yet Unknown

```
switch (n)
{
    case constant1:
        // code to be executed if n is equal to constant1;
        break;

    case constant2:
        // code to be executed if n is equal to constant2;
        break;

        .

        .

        .

    default:
        // code to be executed if n doesn't match any constant
}
```
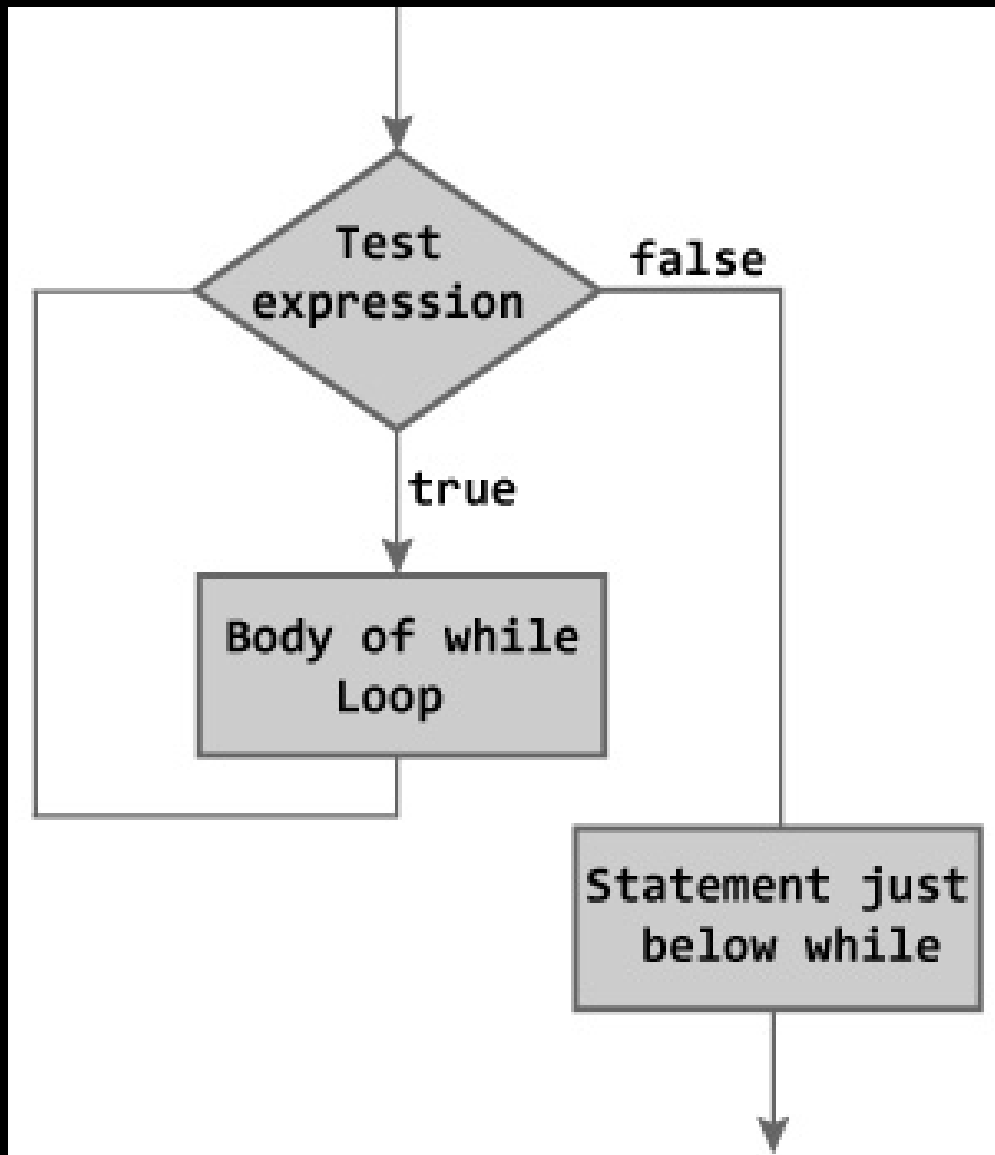
Note the breaks!

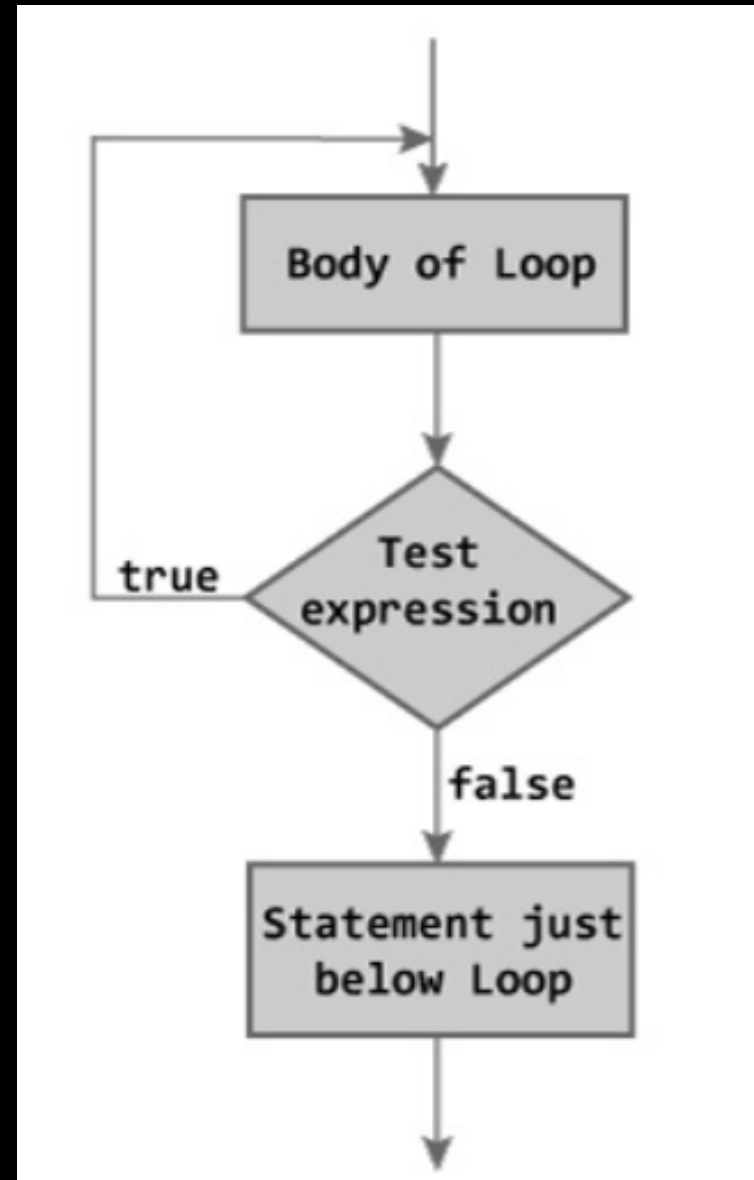# Predictable yet Unknown

# C you again – While loops

```c
while(testExpression) {
    // body of while loop
}


do{
    // body of do-while loop
}while(testExpression)
```

# C you again – While loops
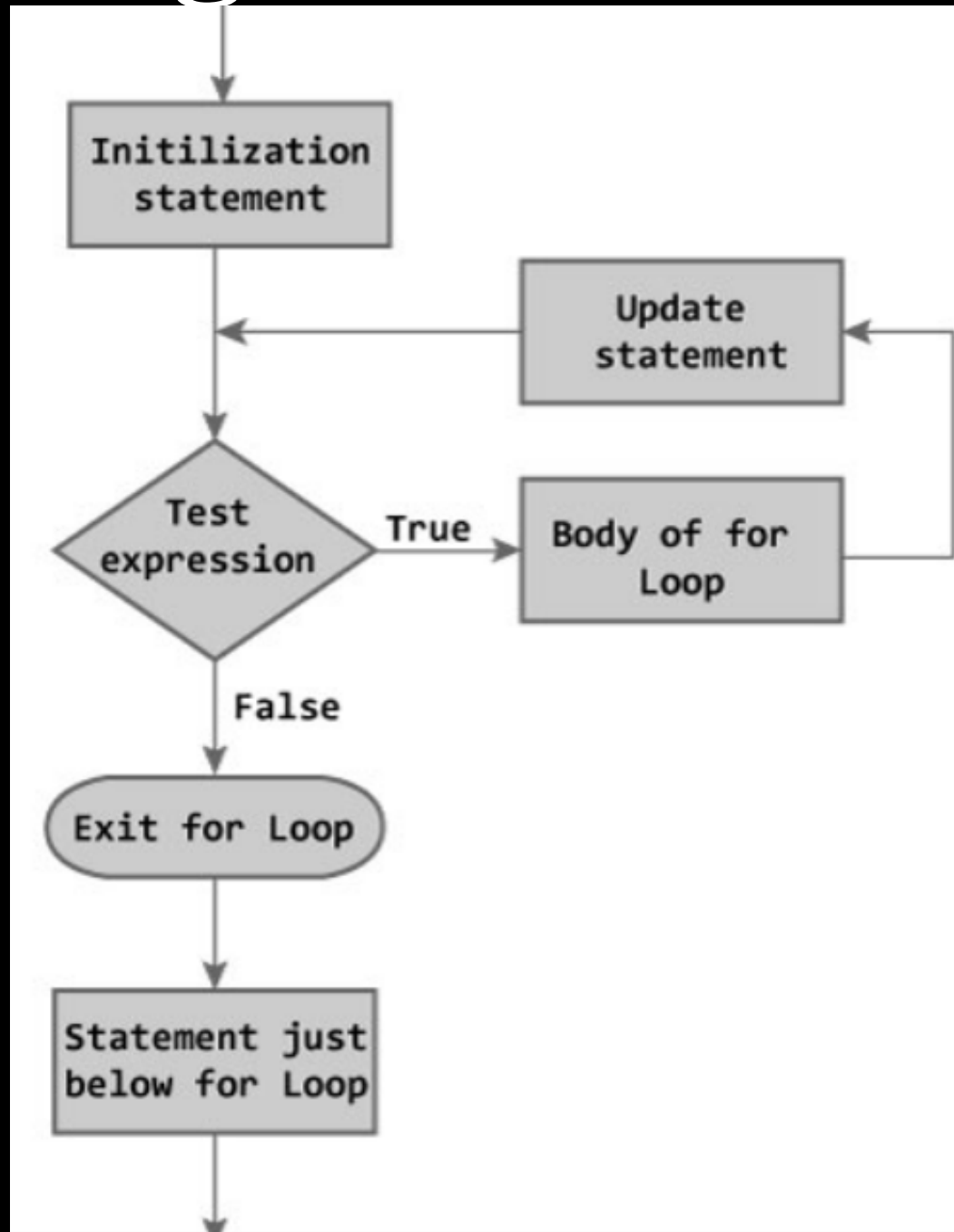


while

do–while

# C you again – For loops

```
for(initializationStatement;
    testExpression;
    updateStatement) {

    // body of for loop

}
```
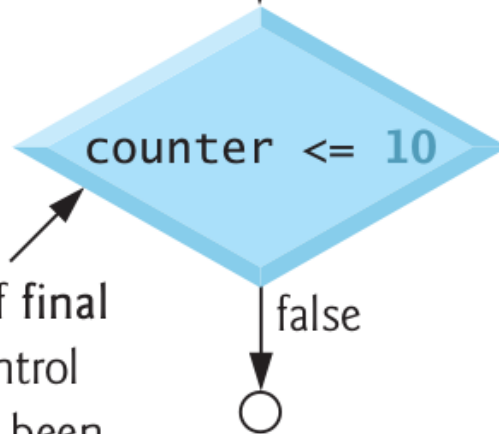
# C you again – For loops

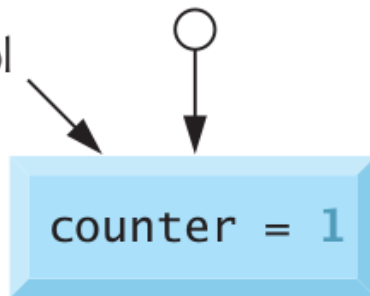# For loop – Example

```
for(counter=1;
    counter<=10;
    ++counter) {


  printf("%u\n", counter);



}
```

# For Loop – Flow



Establish **initial value** of control variable → counter = 1

counter <= 10

Determine if **final value** of control variable has been reached

true → printf( "%u", counter );
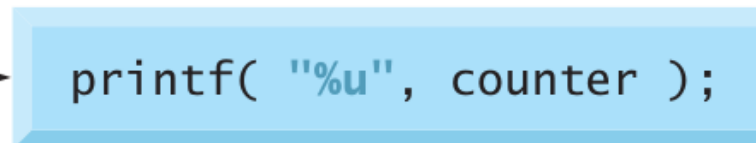
Body of loop (this may be many statements)

++counter
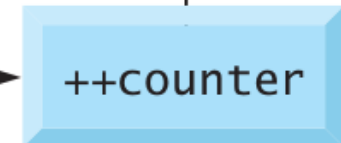
Increment the control variable

false

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = 0; i < n; i++) ...
```

Guess what the above loop counts?

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = 0; i < n; i++) ...
```

Counting up from 0 to n-1

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = 1; i <= n; i++) ...
```

Guess what the above loop counts?

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = 1; i <= n; i++) ...
```

Counting up from 1 to n

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = n-1; i >= 0; i--) ...
```

Guess what the above loop counts?

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = n-1; i >= 0; i--) ...
```

Counting down from n-1 to 0

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = n; i > 0; i--) ...
```

Guess what the above loop counts?

# For loop - Idioms

For loop is the best choice to count up or down!!

```
for(i = n; i > 0; i--) ...
```

Counting down from n to 1

# For loop - Idioms

```
for(i = 0; i < n; i++) ...

for(i = 1; i <= n; i++) ...

for(i = n-1; i >= 0; i--) ...

for(i = n; i > 0; i--) ...
```

Test expressions are crucial to avoid off-by-one error the most common error beginning C programmers make!!

# Did You Notice?

loops often follow the same structure

⭐ Do something simple before the loop, like set a counter.

⭐ Have a simple test condition on the loop.

⭐ Do something at the end of a loop, like update a counter.

# Did You Notice?
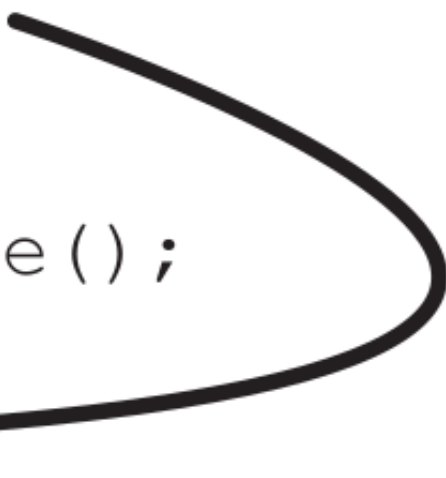
for and while loops are interchangeable

```
for(initializationStatement;
    testExpression;
    updateStatement) {
  // body of for loop
}



                    intializationStatement;
                    while(testExpression) {
                      // body of while loop
                      updateStatement;
                    }
```

# Use break to Break Out!
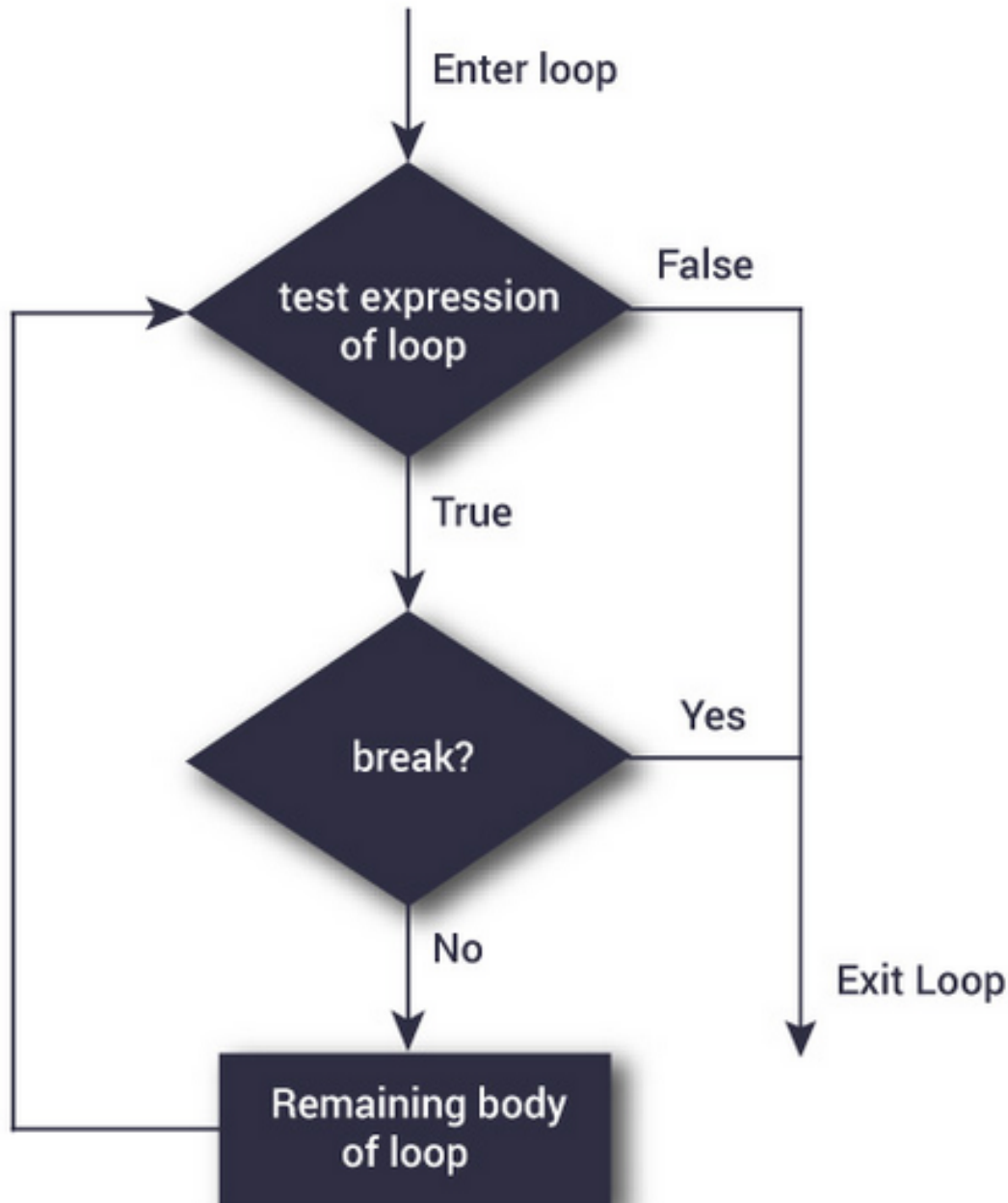
```
while(feeling_hungry) {

    eat_cake();

    if (feeling_queasy) {

        /* Break out of the while loop */

        break;

    }

    drink_coffee();

}
```

"break" skips out of the loop immediately.

breaks don't break if statements!
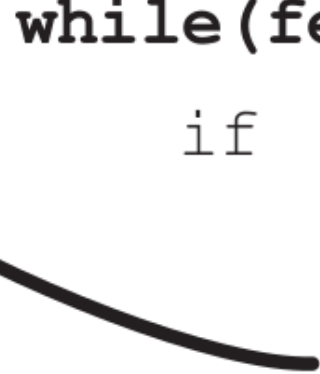
# Use break to Break Out!



```
while (test Expression)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for break)
    {
        break;
    }
    // codes
}
```

Flowchart labels: Enter loop → test expression of loop → False → Exit Loop; True → break? → Yes → Exit Loop; No → Remaining body of loop

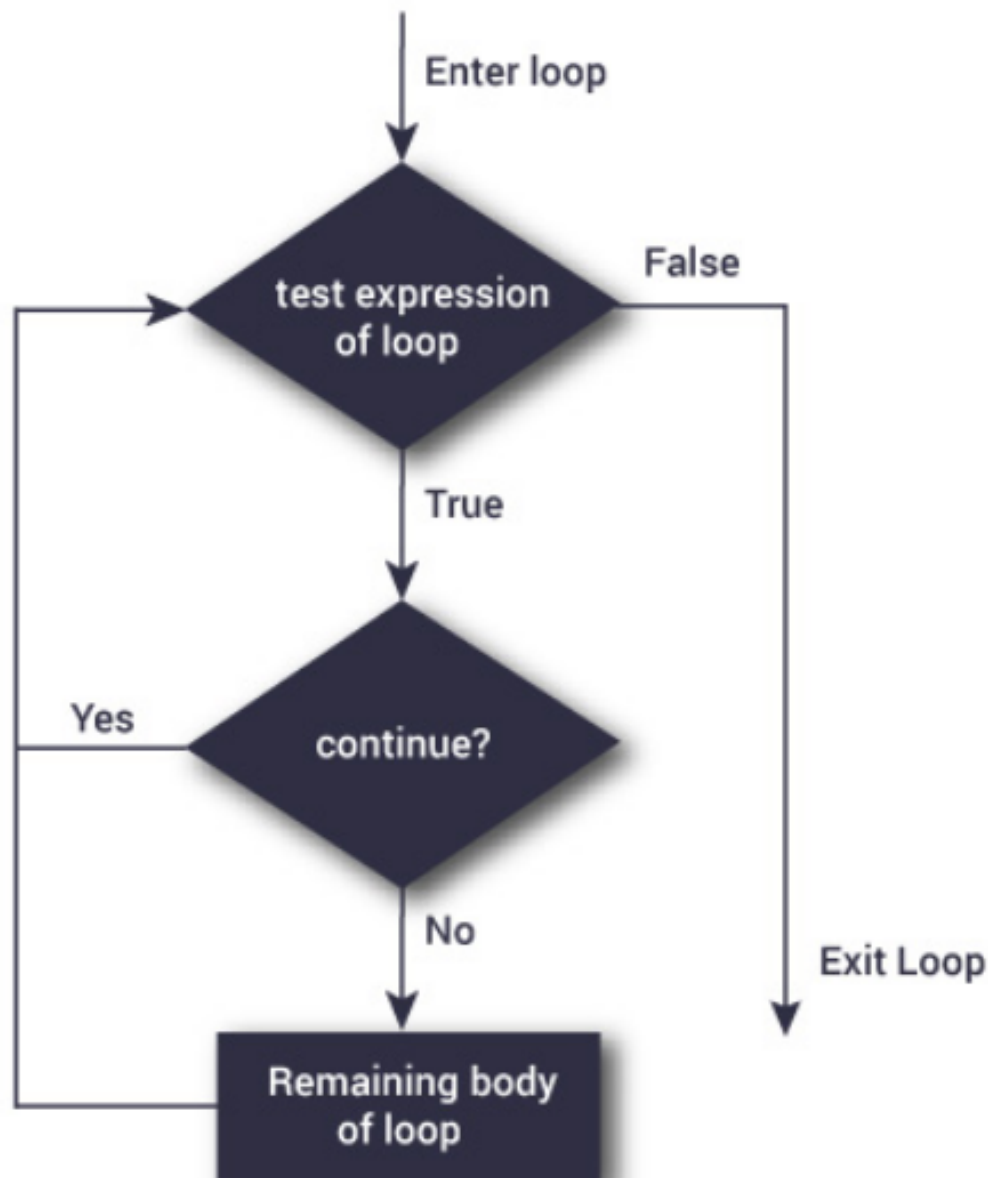# And `continue` to Continue!!

```c
while(feeling_hungry) {
    if (not_lunch_yet) {
        /* Go back to the loop condition */
        continue;    "continue" takes you back
                      to the start of the loop.
    }
    eat_cake();
}
```

# And `continue` to Continue!!



```
while (test Expression)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

Flowchart labels:

Enter loop → test expression of loop → False → Exit Loop
True → continue? → Yes → (back to test expression)
No → Remaining body of loop

# CSE102
# Computer Programming

## (Next Topic)