CS101C: Introduction to Programming (Using C)

Autumn 2025

Nikhil Hegde
Achyut Mani Tripathi

Week4: Loops (while, do-while, for)

So far..

- Printing on and accepting input the terminal (printf, scanf)
- Data Types (int, float, double, char), constants, variables and their initialization using constants
- How much space is reserved in memory for data types and bit representation of data types
- Operators
- Control flow with if, if-else, else-if, switch

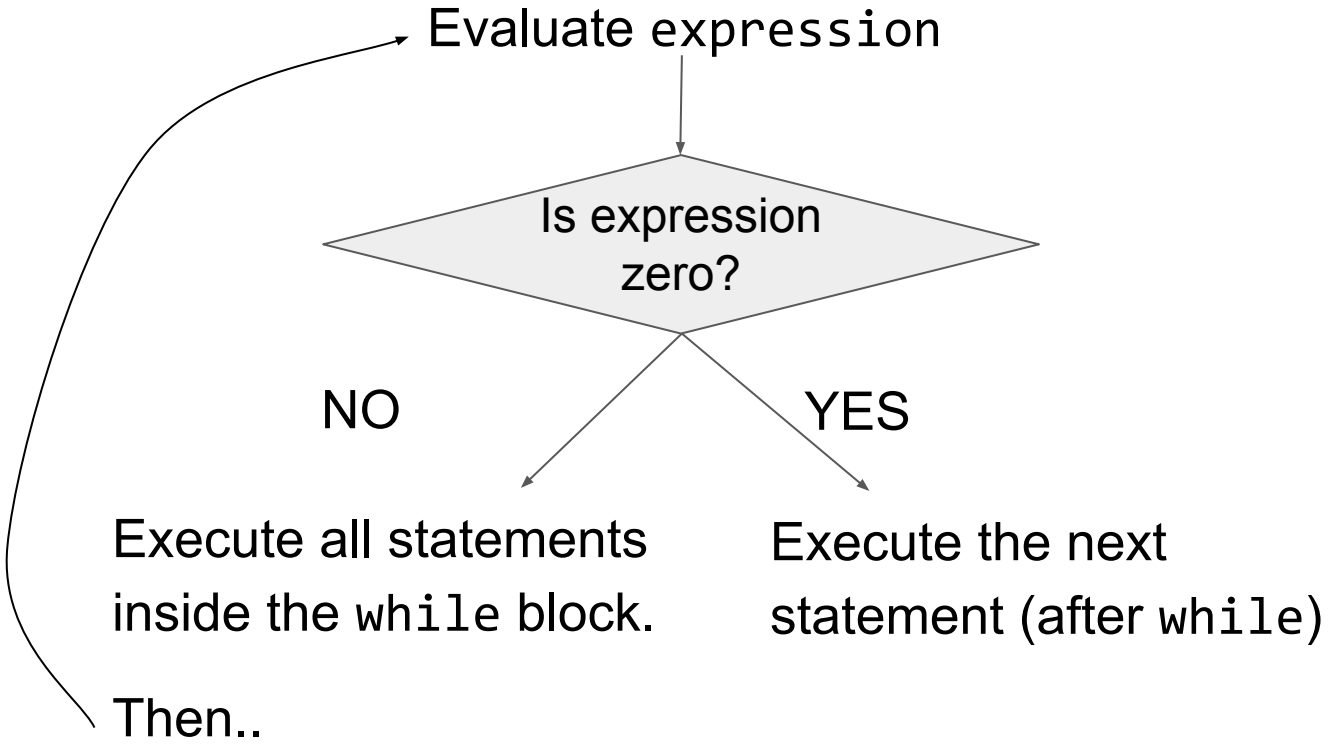
Today’s class (25/8/2025)

- Loops
  - While

`while` statement - syntax and meaning

```
while(expression){
    statement1;
    ..
    statementn;
}
```

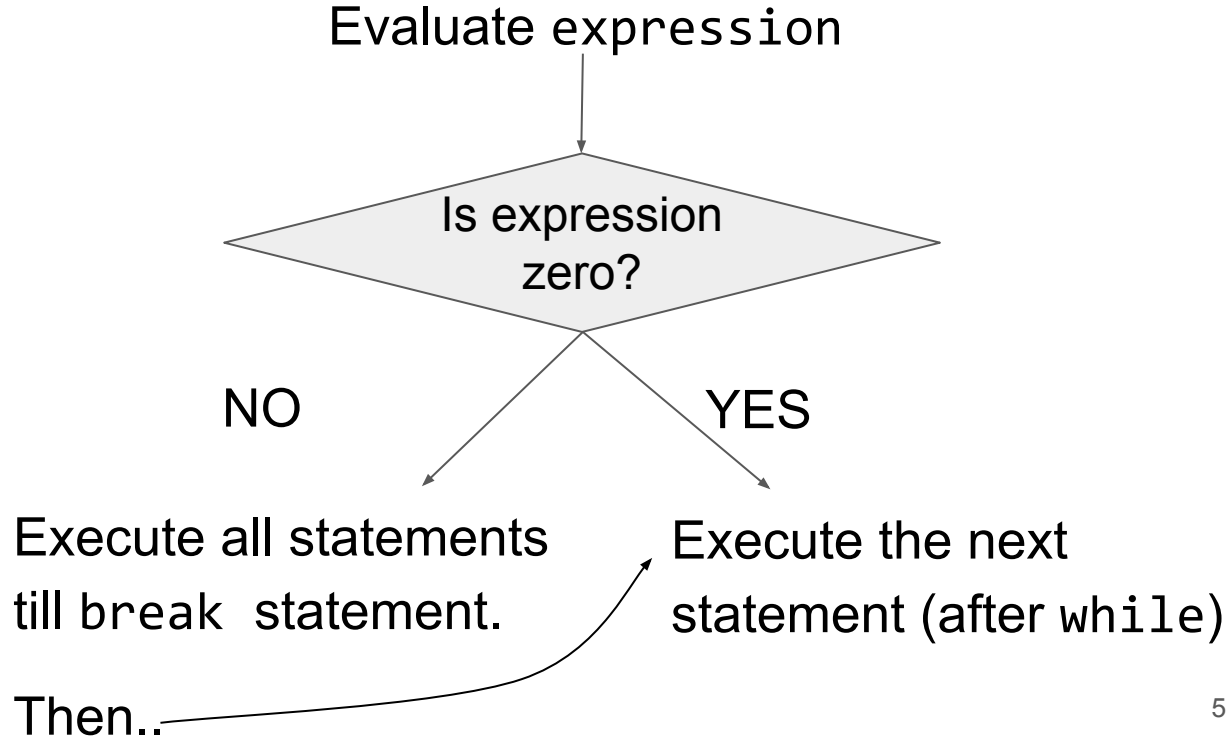
Loop body



`while` statement - syntax and meaning

- The **`break`** statement interrupts this flow.

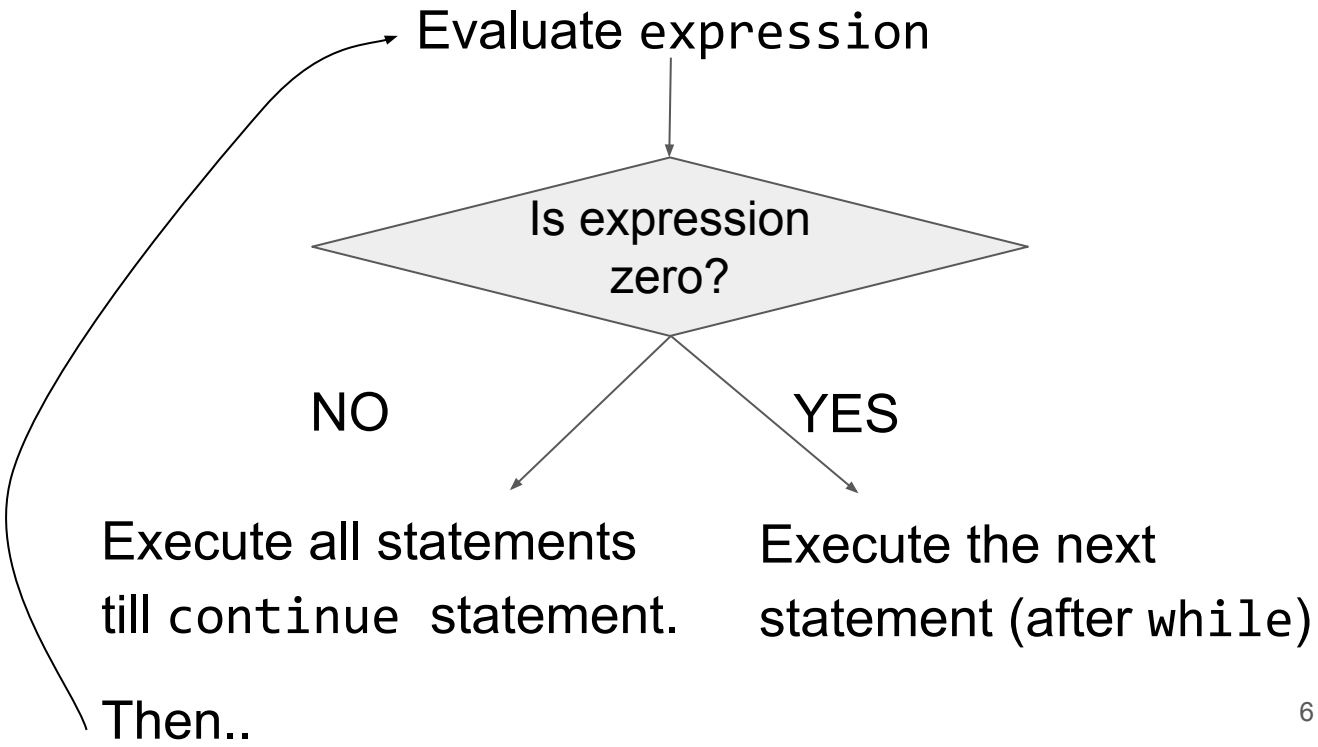
```
while(expression){
    statement1;
    ..
    break;
    statementn;
}
```



`while` statement - syntax and meaning

- The **`continue`** statement interrupts this flow.

```
while(expression){
    statement1;
    ..
    continue;
    statementn;
}
```



Demo - `while` statement

- Write a C program:

  - to accept an integer from terminal.

  - print multiples (1 to 10) of the integer using `while`

Sample Program - `while` statement

```
//program to print multiples of an integer that is input.
int main(){
        int digit;
        int i=1;
        printf("Enter a digit\n");
        scanf("%d",&digit);
        while(i<=10){
                printf(" %d times %d is %d\n",digit, i, digit*i);
                i+=1;
        }
        printf("end of program\n");
}
```

Recap: demo program - `while`

- Write a C program:

  - to accept a 4-digit integer from terminal.

  - print the place value of each digit

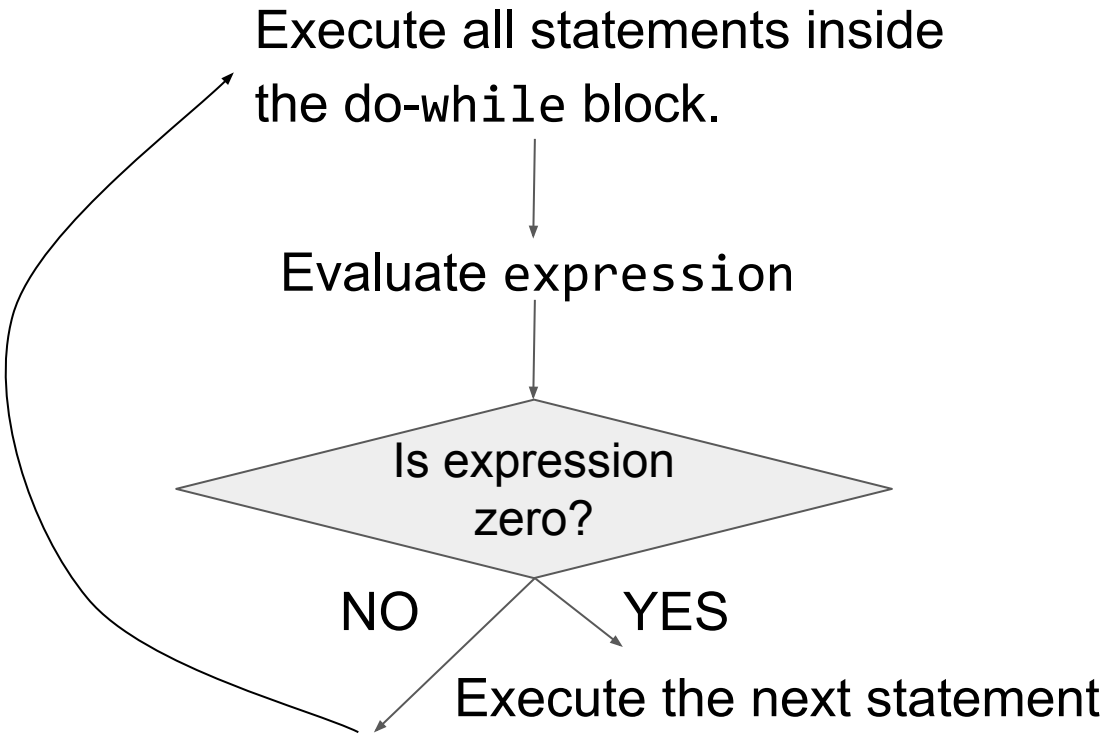
Today’s class (29/8/2025)

- Loops
  - Do-While
  - For

do-while statement - syntax and meaning

```
do{
    statement1;
    ..
    statementn;
}while(expression);
```

do-while block /
Loop body



for Statement - Syntax

```
for(expression1;expression2;expression3) {

statement1;

..

statementn;

}
```

- Expressions can be omitted. Semicolons must remain.
- Curly braces can be omitted when single statement present.
- Most commonly:
  - Expressions 1 and 3 are assignments
  - Expression2 is a relational expression

for Statement - meaning

```
for(expression1;expression2;expression3) {

statement1;

..

statementn;

}
```

1. Evaluate expression1.
2. Is expression2 true? (if expression2 not present, always true)
   a. If false execute the next statement after the for statement
3. Execute statements 1 to n.
4. Evaluate expression3.
5. Go to step 2.

Demo - while loop with comma operator

```c
#include <stdio.h>
int main()
{
    int i = 0, j = 0;
    while (i<5,j<10)
    {
        i++;
        j++;
    }
    printf("%d %d", i, j);
}
```

Demo - for loop

```c
#include <stdio.h>
int main()
{
    short i;
    for (i = 1; i> 0; i++)
        printf("%d\n", i);
}
```

Demo - nested for loop

```
void main()
{
    int i = 0, j = 0;
    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 1;)
        {
            break;
        }
        printf(" Bye \n");
    }
}
```

Demo - for loop with double type

```c
#include <stdio.h>
void main()
{
    double k = 0;
    for (k = 0.0; k < 3.0; k++);
    printf("%lf", k);
}
```

Demo - infinite loop with `for`

```c
#include <stdio.h>
int main()
{
    for (5; 2; 2)
        printf("Hello\n");
    return 0;
}
```

Challenge: What is the equivalent infinite loop with `while`?