CmpE 150 - Week 4

Section - 04

Recap

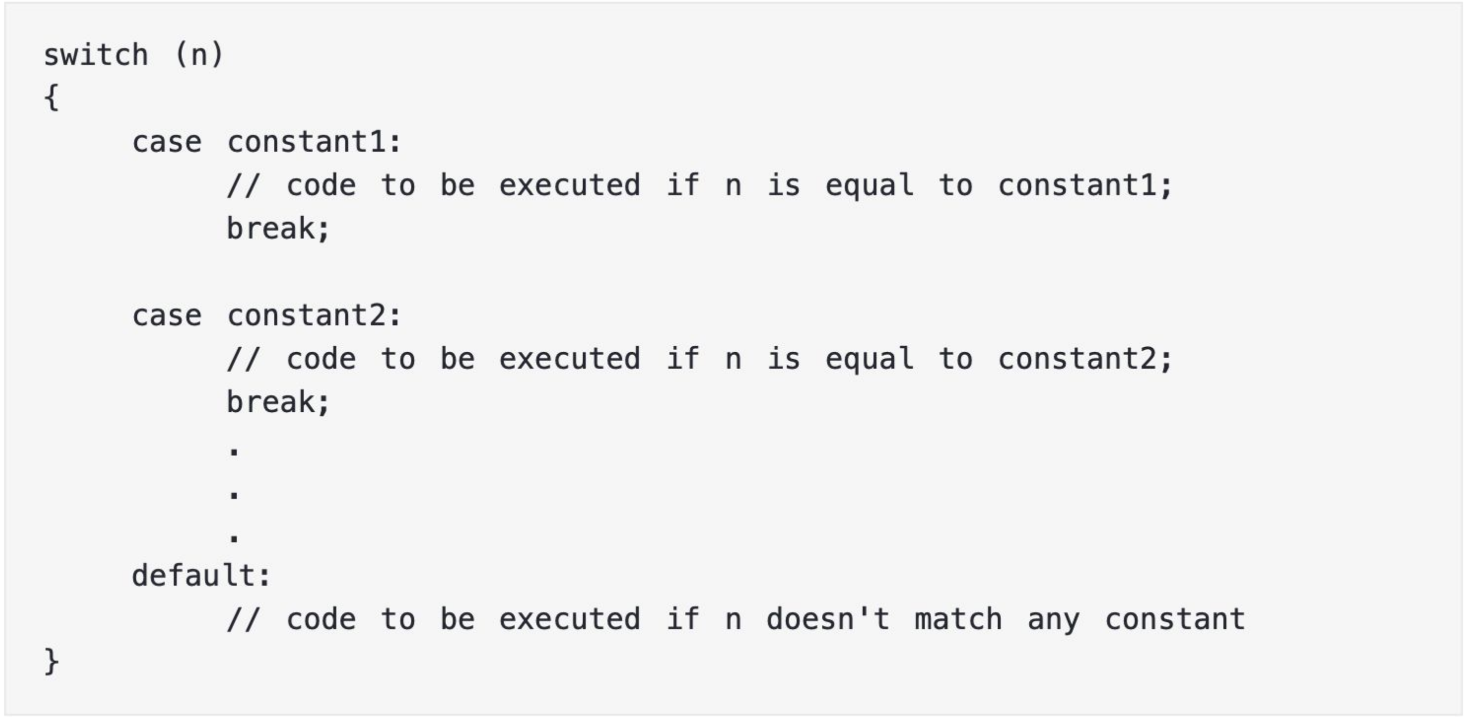
- #define preprocessor directive
- Pre/post-increment operators (++num, num++)
- Selection structures (if, else, else if)

Remark: Slides and some codes shared weekly on:
https://github.com/melsener/cmpe150

2nd quiz today

- Start: 12:00
- End: 12:20
- Be careful! **Do not** write extra things other than required, otherwise automatic grading **does not** give your points!
- No cell phones, no personal computers!

Switch

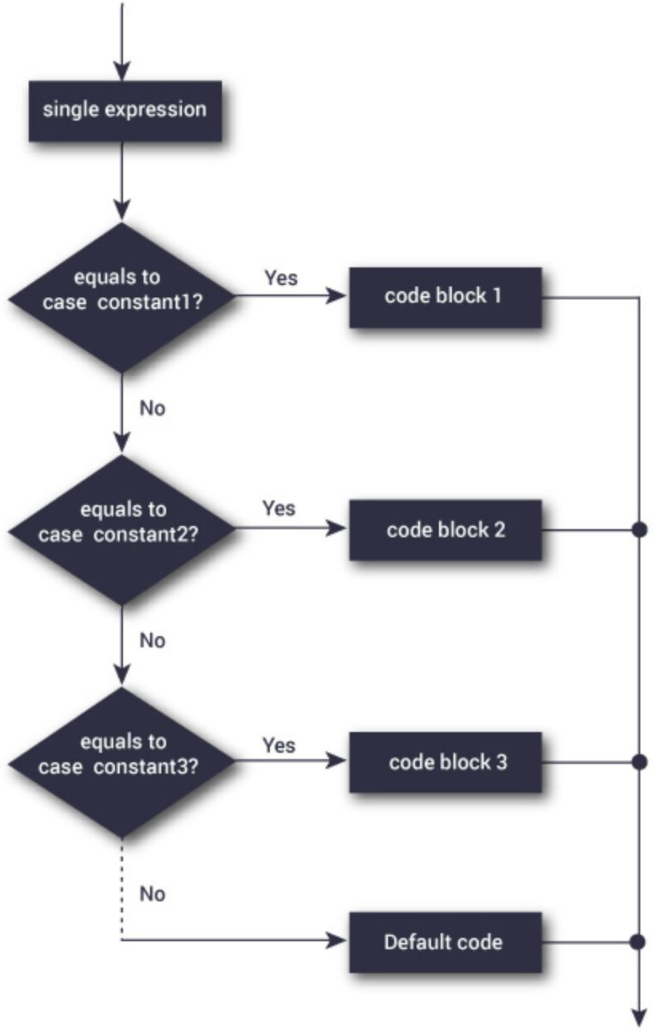


```
switch (n)
{
    case constant1:
        // code to be executed if n is equal to constant1;
        break;

    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
}
```

Source:Programiz - C switch...case Statement

Switch



Source:Programiz - C switch...case Statement

Repetition Structures

A loop is a group of instructions the computer executes repeatedly while some loop-continuation condition remains true.

1. While
2. For
3. Do..while

Src: Deitel C How to Program

While Loop Syntax

```
while(condition) {
    statement(s);
}
```

For Loop Syntax

```
for ( init; condition; increment ) {
   statement(s);
}
```

For Loop Syntax

- The **init** step is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here, as long as a semicolon appears.
- Next, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement just after the 'for' loop.
- After the body of the 'for' loop executes, the flow of control jumps back up to the **increment** statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.

Source: Tutorialspoint - for loop in C