Expressions again

- **Expression**: something that has a value
- **Types of expressions we know**
  - Literal constants: 33 or 3.14
  - Variables: count
  - Simple - two *operands* and *operator*: 3 + 5
  - Complex: (count - (44 - 12) / 7) * num
- **Some expressions have side effect**
  
  ```
  x = 0 /* = is an operator! */
  printf("Hello, World!\n")
  ```
From expressions to statements

- Statement: *expression with a semicolon*
  
  33;
  
  3+5;
  
  x = 0;
  
  x = y = 0; /* x = (y = 0); */
  
  printf("Hello, World!\n");

- A statement makes sense if an expression in the statement has some side effect

Block and sequential execution

- Block: { ..... }
  
  - A group of statements
  
  - Statements are *sequentially* executed
  
  - Syntactically equivalent to a statement

- Example:

  ```
  {
      a = a +1;
      b = a % 2;
  }
  ```
Block and sequential execution

- Flowcharts are used to show the control flow inside the program.
- Sequential execution inside a block means that the control (over the processor) flows downwards from statement to next statement.

```
a = a + 1
b = a % 2
```

While loop

```java
while (expression)
    loopstatement
nextstatement
```

- If expression is not 0 (true) - dive into the loop
- If expression is 0 (false) - skip to nextstatement
- I.e, while expression is true, do the loop
While loop

while (expression) {
    statement-1
    ...
    statement-K
}
nextstatement
- If expression is not 0 - dive into the loop
- If expression is 0 - skip to nextstatement
- I.e, while expression is not 0, do the loop

Flowchart of the while loop

Expression is equal to 0?

Yes

No

statement-1

statement-K

nextstatement
Example: Countdown (1)

/* Example 1: counting to zero  
   Course IS0012    
   Author: Peter Brusilovsky */
#include <stdio.h>
void main()
{
    int counter = 5; /* setting the counter */
    printf("Start counting...
");
    while (counter) {
        printf("%d
", counter);
        counter = counter - 1;
    }
    printf("Fire!
");
}

Increment expressions

- **Post-Increment: num++**
  - Side effect: num is incremented
    - same as num = num + 1
  - Value: the value before increment
    - same as num
- **Pre-Increment: ++num**
  - Side effect: num is incremented
    - same as num = num + 1
  - Value: the value after increment
    - same as num + 1
Decrement expressions

- **Post-Decrement: num--**
  - Side effect: num is decremented
    - same as num = num - 1
  - Value: the value before decrement
    - same as num

- **Pre-Decrement: --num**
  - Side effect: num is decremented
    - same as num = num - 1
  - Value: the value after decrement
    - same as num - 1

Some new operations

- **Special assignment expressions**
  - `result = result + 100;` → `result += 100;`
  - `result = result - 100;` → `result -= 100;`
  - `result = result % 100;` → `result %= 100;`
  - `result = result * 100;` → `result *= 100;`
  - `result = result / 100;` → `result /= 100;`

- **As every expression it has a value**
  - The value after assignment

- **The side effect is the assignment**
Example: Countdown (2)

    /* Example 2 - counting to zero
       Author: Peter Brusilovsky 9/12/00 */
    #include <stdio.h>
    #define HOW_MANY 5

    void main()
    {
        int counter = HOW_MANY; /* setting the counter */

        printf("Start counting...
");
        while (counter)
            printf("%d
", counter--);

        printf("Fire!
");
    }

Example: Interest over Years

    void main() {
        int years; /* years the capital stays in bank */
        float interest_rate; /* interest rate in percent */
        float capital; /* capital in dollars */

        printf("Startup capital ($$$): ");
        scanf("%f", &capital);
        printf("Interest rate in percent (xx.xx): ");
        scanf("%f", &interest_rate);
        printf("How many years? ");
        scanf("%d", &years);

        while (years) {
            capital += capital * interest_rate / 100;
            --years;
        }

        printf("New capital %9.2f
", capital);
    }
Before Next Lecture:

- Do reading assignment
  - Perry: Chapter 10; Chapter 14 (First reading)
- Run Classroom Examples
- Check yourself by working with KnowledgeTree and WADEIn system
- Last HW before the Midterm
- Start thinking about Midterm