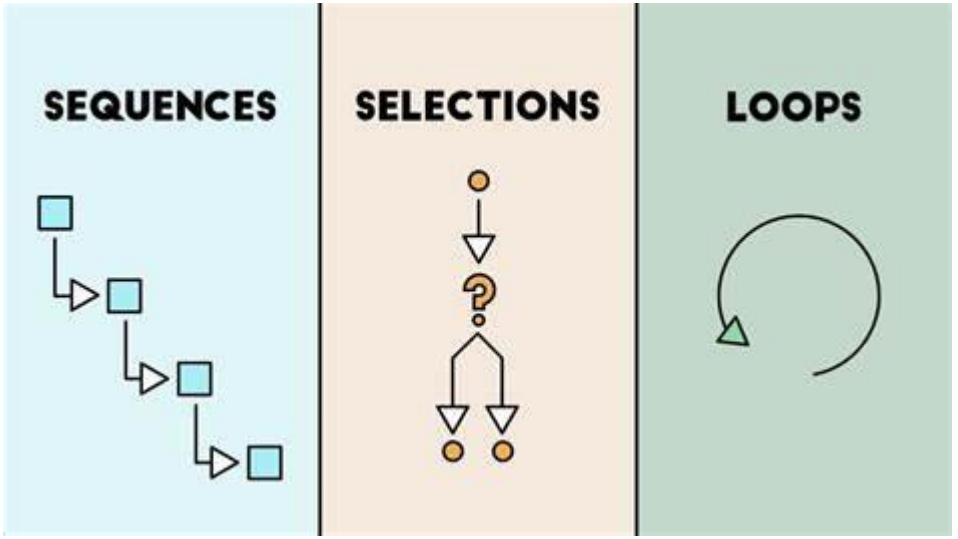
Programming Languages and Tools: Programming with C++ CS:3210:0003

Lecture/Lab #7

while Loops



- For repeated execution
  - Use functions
  - Use loops when there is a common pattern of change in each execution
- Syntax:

```
while(expression) {
   //if expression evaluates to true
   StatementBlock;
}
```

- Repeats as long as expression evaluates to true
- Need to make expression false at some point within the body to avoid infinite loop

Program to count from 0 to 10

Sequential vs Loop

Activity

Write a program countfromn.cpp that inputs a non-negative integer n and prints in descending order every integer from n to 0

Activity

Write a program counttoneven.cpp that inputs a non-negative integer n and prints every even integer from 0 to n

Activity

Update countton.cpp so that the output is printed in 3 columns. Use \t (for tab) within strings to align text in columns. Ex: for n == 7,

```
0	1	2
3	4	5
6	7
```

Activity

Update int2bin.cpp as follows:

- Use a while loop for the calculation and the printing of the binary value in inttobin

Compilation Order

1. Preprocessor Directives: things that begin with #.
   Ex: #include<iostream>
2. Global variables declared before main()
3. Function headers of functions defined before main()
4. main() function
5. Bodies of any functions called from main()

Function Declarations

- aka function prototypes
- Tell the compiler the behavior of the function
- Syntax:
  returnType functionName (typedArgList);
- Function definition = declaration + { body of function }
- Ordering functions to avoid prototypes is not possible when:
  1. Definitions are in other files
  2. Functions call each other
- Declare all your functions in the beginning (except main)

Default Values for Function Arguments

- You can give default values to the arguments of your function in the prototype
- A function call can either use the default value or override it
- Syntax:
  funcType funcName (argType = defValue, …);