**Syllabus**

Introduction to Computing: 89-110
Meir Kalech

**Introduction:**
1. Overview.
2. What is an Algorithm?

**Data Types:**
1. Primitive types.
2. Initializing.
3. Const.
4. Casting.
5. `operator=`.

**Input/Output**
1. `printf`.
2. `scanf`.

**Operators:**
1. Arithmetic operators.
2. `operator++` and `operator--`.
3. Logic operators.
4. Relational operators.
5. Preferences.

**Control flows:**
1. Condition statements.
2. Loop statements.

**Functions:**
1. Declaration, Definition and Function Call.
2. Parameters (call by value, call by address).
3. Return value.
4. Memory stack.

**Variable Scope and Life:**
5. Local variable.
7. Static variable.

**Arrays and Strings:**
1. Array.
4. Strings.
5. string library.
6. Array and 2-dim array as parameter.
7. Call by value, call by address.

**Pointers:**
1. Addresses and pointers.
2. Pointers to arrays.
3. Address operator (\&).
4. Indirection operator (*).
5. Pointer as parameter to function.
6. Pointers and arrays.
7. Array of pointers.
8. Pointer to pointer.

**Dynamic Allocation Memory (DAM):**
1. What is DAM?
2. DAM and pointers.
3. Memory free.
4. DAM scope and life.
5. Pointer to pointer memory allocation.
6. Pointer as parameter to function for allocation.

**Structures:**
1. Declaration.
2. Initialization.
3. Array of Structures.
4. Pointer to structure (operator ->).
5. DAM of structures.

**Recursion**
1. Compare iterative function to recursion.
2. Statement before and after recursion call.

**Precompilation**
1. operator #.
2. macros.
3. Conditional compilation.

**Bits operators**

**Appendices:**
1. Parameters to main.
2. Pointer to function.
3. Const pointer to a structure as parameter to function.