# Programming Techniques — C++ Course Outline (2017/18)

## Lecture 1_a

- Basic structure of a C++ program
- Namespaces and the standard C++ library
- Input and output streams
- Keywords
- Input stream buffer and buffer control

## Lecture 1_b

- Specifics of variable definitions in C++
- Type conversions
- Functions for working with floating-point numbers
- Limitations of numeric data ranges
- Classical, constructor, and uniform variable initialization
- Treatment of constant values
- Formatted output and I/O manipulators

## Lecture 2_a

- Boolean data type and logic expressions
- Enumerated types
- Types and functions for working with complex numbers

## Lecture 2_b

- Vectors as an alternative to arrays
- Vector manipulations
- Initialization lists
- Relationship between vectors and pointer arithmetic
- Passing vectors to and returning vectors from functions
- Range-based for loops
- Automatic type deduction during initialization
- Accessing vector elements with index validation
- Move semantics
- Deques

## Lecture 3_a

- Matrices as vectors of vectors
- Character types in C++
- Manipulation of C-style null-terminated strings

## Lecture 3_b

- Dynamic strings
- Manipulating dynamic strings
- Throwing and catching exceptions
- Standard exceptions

## Lecture 4_a

- References
- Passing parameters by reference
- Using references in range-based for loops
- Input, output, and in-out parameters
- Aliasing issues
- Returning references from functions
- References to constant objects
- R-value references

## Lecture 4_b

- Default function parameters
- Function overloading by number and type of parameters
- Template functions
- Generic functions
- Specialization of generic functions

## Lecture 5_a

- Full and partial type deduction in generic functions
- Concepts and concept models
- Automatic type deduction of expressions
- Alternative function return type specification
- Iterators and their usage
- Basic standard library algorithms
- POD and non-POD types

## Lecture 5_b

- Passing functions as parameters
- Lambda functions
- Function pointers
- Arrays of function pointers
- Standard algorithms using function parameters
- Sorting and searching algorithms

## Lecture 6_a

- Dynamic allocation of individual variables
- Memory management strategies
- Memory leaks and dangling pointers
- Dynamic allocation of one-dimensional arrays
- Exception handling during dynamic allocation

## Lecture 6_b

- Pointers to arrays, arrays of pointers, and double pointers
- Dynamic allocation of multi-dimensional arrays
- Continuous and fragmented allocation
- References to pointers
- Exception handling for multi-dimensional dynamic allocation

## Lecture 7_a

- Specifics of character pointer arrays
- Using pointer arrays for manipulating large objects
- Detailed discussion of memory leak problems
- Shared smart pointers and their usage

## Lecture 7_b

- Unique smart pointers
- Iterators as generalized pointers
- Iterator categories: input, output, forward, bidirectional, random access
- Lists and singly linked lists
- Sets and multisets
- Reverse iterators and inserters
- Stack, queue, and priority queue
- Capturing variables in lambdas and lambda closures
- Polymorphic function wrappers
- Basic ideas of functional programming

## Lecture 8_a

- Structured data types and variables
- Manipulating structured data
- Pointers to structured data
- Dynamic allocation of structured data

## Lecture 8_b

- Generic structures
- Pairs and maps
- Structures with pointer members
- Shallow vs. deep copies
- Nodes and singly linked list implementation
- Circular references with smart pointers

## Lecture 9_a

- Limitations of structures and procedural programming
- Introduction to object-oriented programming
- Classes as an extension of structures
- Attributes and member functions
- Public interface of a class
- Information hiding and encapsulation
- Accessor (getter) and mutator (setter) methods
- Inline functions
- Inspectors and mutators

## Lecture 9_b

- Static attributes and methods
- Examples of object-oriented development

- Friend functions and classes

### Lecture 10_a

- Constructors
- Role of constructor parameters
- Creating anonymous temporary objects
- Copy constructor
- Aggregate initialization
- Implicit type conversion via constructors

### Lecture 10_b

- Direct vs. copy initialization
- Arrays and vectors of class instances
- Arrays and vectors of pointers to class instances
- Constructor initializer lists
- Pointers and references as class members

### Lecture 11_a

- Dynamic memory allocation in constructors
- Resource management automation (RAII)
- Destructors
- Explicit constructors
- Sequence constructors (initializer list constructors)
- Destructor issues and shallow copy (rule of three)
- Deep copy via copy constructor
- Copy assignment operator
- Move constructor and move assignment operator
- Move semantics
- Automatically generated class members

### Lecture 11_b

- Solving destructor/shallow copy issues using reference counting
- Copy-on-write
- Example: creating a container type
- Preventing copy and assignment
- Encapsulating dynamic memory management

### Lecture 12_a

- Operator functions
- Overloading operators via global functions
- Specifics of overloading input/output operators
- Best practices and examples for operator overloading

### Lecture 12_b

- Overloading via member operator functions
- Indexing operator overloading
- Functors (function objects)

- Standard library functors
- Lambda closures as function objects
- Binders, adapters, and holders
- Type conversion operator functions

## Lecture 13_a

- Object-oriented methodology
- Inheritance as a key OOP principle
- Base and derived classes
- Regular vs. irregular inheritance
- Inheritance vs. aggregation
- Function overriding
- Public and private inheritance
- Static vs. dynamic types of pointers and objects
- Early vs. late binding
- Virtual member functions
- Polymorphism and polymorphic variables
- Upcasting and downcasting
- Static and dynamic casting

## Lecture 13_b

- Heterogeneous container objects
- Polymorphism and virtual function usage
- Class hierarchies
- Polymorphic copying
- Surrogate classes

## Lecture 14_a

- Files in C++
- File stream operations
- Working with text files
- Memory buffer-based streams

## Lecture 14_b

- Working with binary files
- Cursor control and direct-access files
- Storing container objects in files
- Serialization