

**School of Computing, Division of Computer Science
Clemson University
CpSc 8700 – OO Software Development with C++
Syllabus, August 27, 2015**

Video Game Topics

1. Introduction
2. Drawing a surface with SDL
3. blitting
4. Data-driven programming with XML
5. L-Systems
6. Colorkeys and transparency
7. Loading image formats
8. Inkscape: vector graphics
9. Sprite animation
10. Sprite clipping
11. Backgrounds & virtual worlds
12. Bouncing sprites
13. Game levels
14. Sound & Music
15. Parallax scrolling
16. Painter's Algorithm
17. Explosions & memory management
18. Artificial intelligence
19. Processing input: mouse, keyboard, joystick
20. Adding sound
21. Collisions, collision detection
22. Particle Systems
23. Game time and timers
24. Adding text
25. Pausing the game

Language Topics

1. **Basics:** I/O, data types, iteration: for, while; control structures: if/else and switch; short-circuit evaluation, functions, The 3 parameter transmission modes, C-strings, command-line parameters, intro to files.
2. **The C++ Class:** Constructors: default, conversion and copy. Class instantiation, constructor initialization vs assignment, when are constructors called, which one! destructors and when they are required. Also, the class members that are supplied by default, those that you **should** supply. Shallow vs deep copy, orthodox canonical class form, functions that C++ silently writes, overloading functions, overloading *operators*, overloadly assignment, the output operator and others. friend functions (functions that are members of a class vs part of a class), nomenclature and programming style, overloading a class for binary arithmetic, make files, dynamic vs static storage, Writing a string class. Deep copy vs shallow copy. Dynamic memory allocation. Comparing C string with C++ strings. iterators, the stack class, the template stack class, exceptions, the linked list class,
3. **The standard library:** Containers: `vector`, `list`, and `map`. Standard library iterators and algorithms for containers. Insertion & deletion.
4. **Inheritance:** What are the kinds of inheritance, what are the kinds of functions that can be involved in inheritance (virtual, purely virtual, non-virtual), public inheritance vs private inheritance, when should inheritance be used, what are the problems with inheritance, casting down the inheritance hierarchy, the 4 kinds of cast in C++, passing parameters to a base class.
5. **Design Patterns:** Model-View-Controller, Singleton, Composite, Factory Method, Flyweight, Observer, Strategy, Command, and Visitor.
6. **Generic Programming:** Template functions and classes. Meta-programming. Policy classes.