NEW COURSE PROPOSAL

GRADUATE Level I _____ Level II _____ **UNDERGRADUATE** X

SCHOOL, DEPARTMENT, COLLEGE: College of Computing **DATE:** Feb 2004

1. Proposed Course Number: CS1316 (Verify with Registrar's Office)	2. Hours: *LECTURE* 3 *LAB/RECITATION* 0 *SEMESTER CREDIT* 3

3. Descriptive Title:
Representing Structure and Behavior

4. Recommended Abbreviation for Transcript – (24 characters including spaces):

R	E	P		S	T	R	U	C	T	U	R	E		&		B	E	H	A	V	I	O	R

5. Catalog Description – (25 words or less)
Modeling the structure of media (e.g., music, graphical scenes) using dynamic data structures. Designing objects as encapsulations of structure and behavior. Algorithms for simulating objects.

6. Basis: L/G X P/F X Audit X

7. Prerequisites: CS1315

Prerequisites with concurrency:

Corequisites:

8. Has the course been taught as a special topic? No If YES, When Enrollment

9. Is this course equivalent to another course (graduate or undergraduate) taught at Ga. Tech? If yes, list course number(s): CS1321, CS1371

10. Are you requesting that this course satisfy: Humanities Social Science

11. Expected Mode of Presentation:	*MODE*	*% of COURSE*
	Lecture	50%
	Laboratory Supervised / Unsupervised	25%
	Discussion	25%
	Seminar	
	Independent Study	
	Library Work	
	Demonstration	
	Other (Specify)	

12. Planned Frequency of Offering:	*TERM TO BE OFFERED*	*EXPECTED ENROLLMENT*
	Fall	60
	Spring	120
	Summer	

13. Probable Instructor(s) – *Please mark with an asterisk any non-tenure track individuals.*

Guzdial, MacIntyre, Stasko, LeBlanc, Leahy*, Sweat*

14. Purpose of Course: Relation to other courses, programs and curricula:
With CS1315, creates an alternative to CS1321 and CS1371, thus allowing a media-centric introduction to computing—especially important for BS in Computational Media. Meets CS1322 pre-requisite.

15. Required X Elective X

16. Please attach a topical outline of the course

Course Objectives:

- Students will be able to construct, modify, and traverse dynamic data structures (e.g., linked lists, trees, and networks) used to represent underlying structure of media (e.g., a network of musical elements traversed to generate a song, or a tree of graphical objects which are traversed to redraw a scene).
- Students will use CRC Cards to analyze situations and design objects for simulating those systems.
- Students will be able to explain the elements of a discrete event simulation.
- Students will be able to explain alternative implementations of an event queue in a discrete event simulation.
- Students will be able to implement an event queue using at least two kinds of sorting mechanisms and explain the tradeoffs of each, including the algorithmic complexity (big Oh) of each.
- Students will be able to implement simple graphical user interfaces using windows, buttons, text areas, and list boxes.

Topical Outline:

1. Introduction to Java from Python: Basic operations (looping, conditionals, assignment), and data types.
2. Media computation in Java: Manipulating pictures, sounds, and text using Java.
3. Object-oriented programming in Java: Classes, instances, and methods.
4. Creating dynamic data structures: Linked lists.
   Students will create, insert, and remove elements of a linked list that represent sound or graphical elements, then replay or redraw by traversing those elements.
5. Creating dynamic data structures: Trees
   Students will create, insert, restructure, and remove elements of a tree that represent sound or graphical elements, then replay or redraw by traversing (both breadth-first and depth-first) those elements.
   Introduce graphical "layering" as differences in ordering of elements.
6. Extending from trees to networks (graphs) with introductions to their general characteristics (e.g., degrees-in and degrees-out, depth) and core algorithms.
7. Objects as representation of structure and behavior.
   Modeling with objects.
   Using object-oriented analysis with CRC Cards to identify structure and behavior of objects.
8. User interface elements as objects.
   Creating and assembling user interfaces from windows, buttons, text areas, and lists.
9. Laying out user interfaces: Layout managers as manipulators of dynamic data structures.
10. Event processing: Connecting the user to the model objects through the user interface objects.
11. Introducing discrete event simulation.
    Elements of a discrete event simulation: Events, event queues, actors, processes, and distributions.
12. Event queue processing: The heart of a discrete event simulation.
    Distributions to create realism and complex behavior
13. Processing events in time order and priority order.
    Forms of sorting to create a sorted list, with their algorithmic complexity.
14. Maintaining a sorted list of events through data structures (e.g., linked lists and trees).
15. Constructing discrete event simulations controlled from graphical user interfaces.