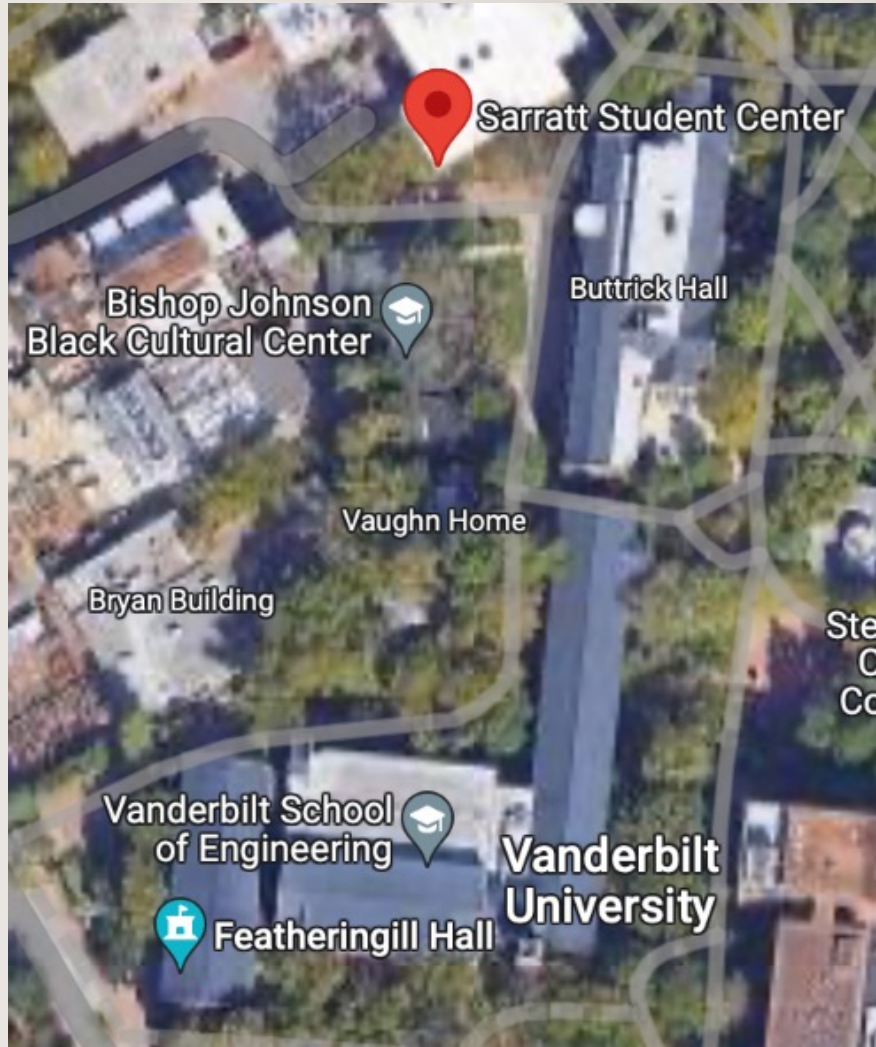
About the Final Exam…

Format

- **Paper-based**, closed book, closed notes
- Short answer, multiple choice, true/false, code reading and writing **(a lot)**
- **Friday, April 28, 7pm – 10pm**
  - **Sarratt Cinema**
  - Arrive early!!!
  - No alternate dates/times provided
- Includes a *reference guide*



Learning Objectives – Chap 1

- General computer science terminology
- Java terminology (e.g., compile, bytecode, JVM, interpreter…)
- Java program structure (e.g., class, method, variable, header…)
- Using a `Scanner` object to get input from the user
  - Prompt for user input first, and then read in input with Scanner
- Output statements using `print`, `println`, and `printf`
- Types of programming errors: syntax, runtime and logic
  - Identify, and describe if the program would compile and run

Learning Objectives – Chap 2

- Java identifiers (e.g., naming convention)
- Declaring, initializing/assigning, and using variables
- Various data types (primitive data types and objects)
- Expressions
- Arithmetic operators and operator precedence
- Type conversion: promotion/coercion and type casting
- `Math` class methods
- Escape sequence

Learning Objectives – Chap 3

- Problem decomposition (e.g., create classes and methods)
- Variable scope
- Declaring methods and calling methods
  - Return type, parameters, data type of the parameters
- Using parameters to pass information to a method
  - Primitive data types – pass by value
  - Objects – pass by reference (its value in the stack is the reference)
- Returning a value from a method

Learning Objectives – Chap 4

- Decision statement structure – `if` statements
- Equality, Relational and Logic operators, and operator precedence
- Short circuit evaluation
- The `boolean` type
- String methods
  - length, equals, equalsIgnoreCase, toLowerCase, toUpperCase, trim, etc…
  - split (split a String into a String array)
- The `char` type and the `Character` wrapper class methods
  - isDigit, isLetter, toLowerCase, toUpperCase, etc…

Learning Objectives – Chap 5

- `while` loops, `do-while` loops, `for` loops
  - Nested loops
  - Fencepost problems, and sentinel values
- `Scanner` next methods
  - nextInt, nextDouble, next, nextLine()
  - next() vs. nextLine() → also commonly used to discard invalid input
- The `Random` object and generating random numbers
- Assertions

Learning Objectives – Chap 6

- File input
  - FileInputStream
  - `Scanner` next and hasNext methods
  - Token-based processing
  - Line-based processing
    - Creating a `Scanner` object on a `String` and tokenize it
- File output
  - FileOutputStream
  - PrintWriter

Learning Objectives – Chap 7

- Array construction
  - Shorthand vs. for loop
  - Arrays of primitive data type
  - Arrays of objects
    - Two-phase initialization: construct the array first, construct each element
    - Use if statement to check if an element is null or not before accessing it
- Array bounds and array indices
  - The `length` field
  - Accessing and processing array elements
- Passing entire arrays to a method
- Returning entire arrays from a method

Learning Objectives – Chap 7

- Array modification
  - Search an element in an array
    - Sequential
    - Binary search
  - Swap array elements
  - Sort an array
    - Selection sort
    - Insertion sort
- The `Arrays` class and its methods (e.g., `fill`, `sort`, `equals`, `toString`, etc…)

Learning Objectives – Chap 9

- Class
  - Instance variables and methods
  - Static fields and methods
  - Constructors
    - Default constructor
    - Overloading constructors
  - Implicit parameter `this`
- Access modifiers → public vs. private vs. protected
  - Encapsulation (information hiding)
  - Accessors and mutators

Learning Objectives – Chap 10

- Why inheritance is useful
- Class hierarchies → Base class (superclass) vs. derived class (subclass)
  - Subclass **extends** superclass
  - ***Is-a*** vs. ***has-a*** relationships
  - Use of keyword `super`
- Polymorphism
  - Declared variable type determines which methods are available
  - Actual type of object determines **which overridden method** gets invoked
- Overriding methods (particularly `toString` and `equals`)
- Overriding vs. overloading

Learning Objectives – Chaps 9 & 10

- Be able to 1) explain, 2) discuss the benefits, and 3) implement code that demonstrate the principles of Object-Oriented Programming
  - Abstraction
  - Encapsulation
  - Inheritance
  - Polymorphism

Exam Prep

Be familiar with the following:

- All programming assignments you have completed
- All demonstrations done in class
- All problems in Midterm Exams 1 & 2
- Understand what was done and why
- Understand how they work

Code Reading

- Read segments of Java code and
  - Determine the output they produce
  - Determine the value of variables
  - Determine the value of expressions
  - Find program errors
  - Trace through control flow constructs

Code Writing

- Write segments of Java code
  - Write simple expressions
  - Declare and initialize variables
  - Write control flow constructs to do conditionals and/or loops
  - Produce the requested output
  - Write a class definition with appropriate instance variables and methods
    - Public vs. private
    - Accessor and mutators
    - Inheritance, derived classes