CS 307

Outline of Material for Test 1

The following is an outline of the material which will be covered on the first test. All of this information can be found in the lecture notes, slides, and/or on this web page.

1. **The Objects of Object Oriented Programming**
    a. A number of terms were defined related to the objects of Object Oriented Programming. Given the definitions of any of the following terms you should be able to give the correct term: **Object, Encapsulation, Data Hiding, Data Abstraction, Inheritance, Multiple Inheritance, Polymorphism, Message (in reference to calling functions), Instance, Instantiation, Interface File, Implementation File, Member access specifiers, Regression Fault, Virtual Function.**
    b. An object, as it relates to software engineering, is a software "bundle" consisting of what two things?
    c. List and briefly explain the three characteristics of an object?
    d. What is a class and how does it relate to an object in a software system?
    e. In the definition of a class (in the .h file) what is the syntax for indicating the class is a sub-class of another class?.
    f. In the definition of a class (in the .h file) what is the syntax for indicating the class has more than one parent class, i.e. has multiple inheritance?.
    g. Briefly discuss the concept of objects interacting by exchanging **messages**

2. **Object Oriented Software Engineering**
    a. A number of terms were defined related to Object Oriented Software Engineering. Given the definitions of any of the following terms you should be able to give the correct term: **Structured/Classical development paradigm, Unified Process (Method), Object Oriented Programming.**
    b. What were the three parts of Modular Programming in the Structured or Classical paradigm approach to software engineering that was primarily used from 1975 to 1985.
    c. In Structured Systems Analysis which is part of the Structured/Classical paradigm what was involved in each of the following techniques: (1) Logical Data Modeling, (2) Data Flow Modeling, (3) Entity Behavior Modeling.
    d. What are the five phases of software development in the Waterfall Method which was an example of the Classical paradigm? What are the other two phases that are sometimes added?
    e. What are the seven steps in the **Unified Process** of Software Engineering developed by Booch, Jacobson, and Rumbaugh?
    f. What are the four phases (iterations) in the **Unified Process** of Software Engineering developed by Booch, Jacobson, and Rumbaugh and what is the primary aim of each?

g. During which of the four phases (iterations) in the **Unified Process** do you do each of the following activities: **(1) Determine if it is worthwhile to develope the target software product, (2) Ensure the client's requirements have been met, (3) Refine the initial requirements, architecture, risks and priorities, redefine the business case, (4) Produce the first operational version (beta release) of the software**?
h. During which Workflow of the **Unified Process** do you usually write each of the following documents: Software Development Plan, Requirements Definition Document, Requirements Specification Document?
i. During which Workflow of the **Unified Process** do you usually decide on what classes will be needed as part of the Architectural Design?
j. During which Workflow of the **Unified Process** do you usually decide on what member variables, and functions will be needed in each class as part of the Detailed Design?
k. We studied seven different examples of Agile methodologies (Extreme Programming-XP, Scrum, Feature Driven Development, Dynamic System Development, Lean Software Development, Kanban Method, and Crystal Family). Identify which methodology each of the following techniques/processes/activities is most associated with:
    a. Frequent releases, short duration cycles
    b. Pair programming
    c. Regular builds, integration testing, and rapid feedback
    d. Sprints
    e. Product Backlog
    f. User Stories
    g. Daily Stand-up Meeting
    h. Two-week work cycle following a 2-step process each time
    i. Defined requirements in the format **<verb>the<something>for/of/to a(n) <something else>**
    j. One of its basic principles is **Eliminate Waste**
    k. Organizes work using a bulletin board divided up into four columns: **Queue, Work-in-Progress, Testing, Live (delivered)**.
    l. Tasks are rated by number of people (color) and sensitivity to error (hardness)

3. **Capability Maturity Model Integrated**
    a. List and briefly define the five levels of capability given in the Capability Maturity Model.
    b. In which of the five levels of the CMMI are each of the following activities a focus: **(1) Basic project management processes are established to track cost, schedule, and functionality, (2) Detailed measures of the software development process and product quality are kept, (3) Continuous process improvement is conducted by using feedback from the software development processes, (4) There are few standards and success depends on individual**

efforts and heroics, (5) The company uses standardized processes for both management and software engineering?

4. **UML**
    a. What are the three main **Parts** of The Unified Modeling Language (UML)?
    b. One of the three Parts that make up the Unified Modeling Language (UML) is **Building Blocks**. What are the three types of **Building Blocks**?
    c. One of the three types of Building Blocks that make up the Unified Modeling Language (UML) is **Things**. What are the four types of **Things**?
    d. Given the name of a Structural thing in UML be able to draw the appropriate graphic.
    e. Given the name of a Behavioral thing in UML be able to draw the appropriate graphic.
    f. Given the name of a Grouping thing in UML be able to draw the appropriate graphic.
    g. Given the name of an Annotational thing in UML be able to draw the appropriate graphic.
    h. Given the name of a Relationship in UML be able to draw the appropriate graphic.

5. **Software Requirements and Design**
    a. A number of terms were defined related to requirements specification and design. Given the definitions of any of the following terms you should be able to give the correct term:**Requirement, Requirement Specification, Requirements Analysis, Step-Wise Refinement, Loose-coupling, Delegation.**
    b. What are the four steps which should be followed in determining the requirements of a system?
    c. What are some good techniques to use when eliciting requirements of a system? We listed 8 different activities.
    d. What are the three parts required in a **Use Case**?
    e. Briefly explain what the following design principle means relative to object oriented software design: ***Identify the aspects of your application that vary and separate them from what stays the same.***
    f. Briefly explain what the following design principle means relative to object oriented software design: ***Program to an interface, not an implementation.***
    g. Briefly explain what the following design principle means relative to object oriented software design: ***Favor composition over inheritance.***
    h. Briefly explain what the following design principle means relative to object oriented software design: ***Strive for loosely coupled designs between objects that interact.***
    i. Briefly explain what the following design principle means relative to object oriented software design: ***Classes should be open for extension but closed for modification.***

6. **Programming Bits and Pieces**
    a. **Variables**

        0. What is an **enumerated data type**? Show how to define one, create a variable of the type, and store values in the variable.
        1. What is a reference variable? Show how to create and use one.
        2. List, and briefly define, the four types of **type-casting** used in C++.
        3. Given examples like: **"Cast the double value 7.9 to 7"** or **"Cast the CShape pointer *sh to a pointer to a COval"** be able to show the correct syntax for the appropriate type of cast, e.g. **int x=static_cast<int>(7.9);** and **COval *co = dynamic_cast<COval *>(sh);** Be able to do this with all four types of cast: **static_cast, reinterpret_cast, const_cast, and dynamic_cast.**

    b. **Functions**

        0. What is **function overloading**? Show how to create an overloaded function in a class.
        1. Explain what **default arguments** to functions are and show how you would declare default values for arguments to a function in the function's prototype.
        2. Show how to define a pointer to a function given the function arguments and return type. Show how to set the function pointer pointing to a function of the appropriate type, and how to use the pointer to call the function.

    c. **Standard Template Library**

        0. Show how you would do the following with each of the most used container templates (vector, list, deque, stack, and queue) of the STL.
            - Instantiate an object of the given type.
            - Insert items into the container. Know which containter templates can have items added at just the front, just the back, or both, and which templates can have items added randomly.
            - Remove/delete items from the container. Know which containter templates can have items removed at just the front, just the back, or both, and which templates can have items removed randomly by using iterators to reference the items.
            - Search/find items in the container if possible.
            - Show how you would scan all items in the container using a forward iterator in a **for** loop.