

Time: May 3, Thursday, 8:00 – 10:30am.

Theory: 20 points

1. What is the “make” utility program?

Answer: Make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program.

2. What are macros in a makefile?

Answer: A makefile can contain definitions of macros. Macros are usually referred to as *variables* when they hold simple string definitions, like "CC=gcc".

3. What is a class diagram?

Answer: a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

4. What is a system sequence diagram?

Answer: a system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events.

5. What is an assertion?

Answer: an assertion statement specifies a condition that you expect to hold true at some particular point in your program.

6. What is an interface file (a.k.a., header file) and what is an implementation file?

Answer: the file that contains the interface is often called the **interface file**, and the file that contains the implementation is called the **implementation file**.

7. What is the problem that `#ifndef` in an interface file solves?

Answer: You can avoid multiple definitions of a class (or anything else) by using `#ifndef` in the header file (interface file).

8. What is a pointer?

Answer: A **pointer** is the memory address of a variable.

9. The link field in the last node in a linked list has a special value stored in it. What is it?

Answer: The value is the *null* pointer. (*Not the NULL pointer.*)

10. What is a driver program (a.k.a., test driver)?

Answer: Drivers are programs that call a function to be tested. Drivers should be simple enough that their correctness is assured.

11. What is a stub?

Answer: A stub is a small function that replaces another as yet untested function. The stub allows the caller to be tested without the added complexity of the untested function.

12. What is a constructor?

Answer: A constructor is a member function that is automatically called when an object of that class is declared.

13. What is a class diagram?

Answer: a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes..

14. What is a namespace?

Answer: A namespace is a collection of names (definitions of constants, variables, functions and classes) that can be made available in a scope by command.

15. What is the problem that the C++ namespace facility solves?

Answer:

This prevents libraries from preempting names and so preventing use of these names by a programmer who is a user of the library. In large program development this facility permits different teams of programmer to use, identical names for different purposes yet avoid conflict.

16. What is a function's signature?

Answer: A function's signature is the function's name with the sequence of types in the parameter list, not including the const keyword and not including the ampersand.

17. What is overloading?

Answer: If you have two or more function definitions for the same function name, that is called overloading.

18. What is a data flow diagram?

Answer: A data flow diagram (DFS) shows the flow of data from external entities into the system. DFD also shows how the data moved from one process to another, as well as its logical storage.

19. What is a use case?

Answer: A use case describes a sequence of actions that provide something of measurable value to an actor.

20. What is a use case diagram?

Answer: Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

Design and Testing: 30 points

1. System sequence diagram

Elements of a system sequence diagram include actors, system events, and information from the system back to the actors (optional).

Given the following simple cash-only process sale scenario, please draw a system sequence diagram for this process sale scenario.

Simple cash-only Process Sale Scenario

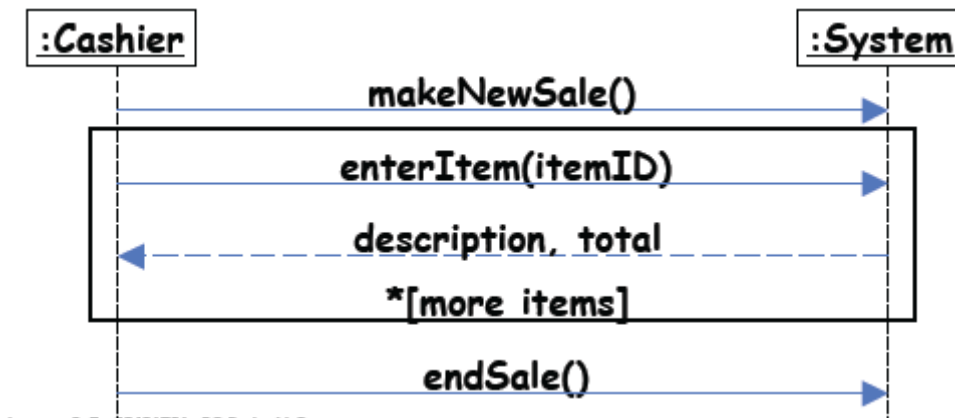
1. Customer arrives at a POS checkout with items to purchase.
 2. Cashier **starts a new sale**.
 3. Cashier enters **item identifier**.
 4. System records sale line item, and presents item description, price and running total.
- Cashier repeats steps 3-4 until done.

(Hint) The following events may be used:

- `makeNewSale()`
- `enterItem(itemID, quantity)`

- `endSale()`

Answer :

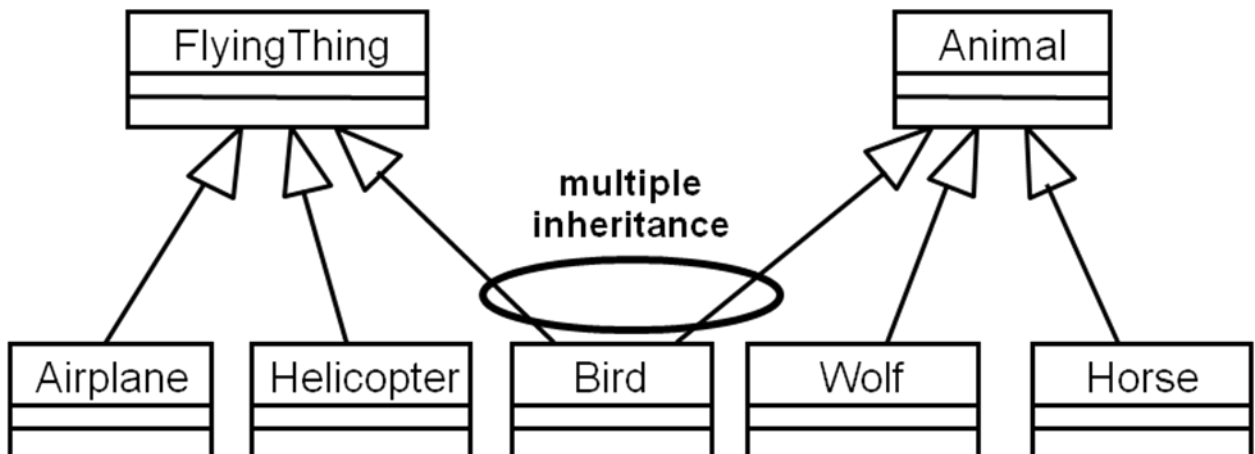


2. Class Diagrams and Design Classes



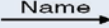
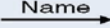




Example: Design a Menu Class

```

class Menu {
public:
    void displayOptions( );
    void addAnOption(string newMessage);
    bool selectOption(int& option); //must be updated to handle errors.
private:
    vector<string> optionMessage;
};
  
```



3. Data Flow Diagram

Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
Every <i>process</i> has A number A name (verb phrase) A description One or more output data flows Usually one or more input data flows	Label (name) Type (process) Description (what is it) Process number Process description (Structured English) Notes		
Every <i>data flow</i> has A name (a noun) A description One or more connections to a process	Label (name) Type (flow) Description Alias (another name) Composition (description of data elements) Notes		
Every <i>data store</i> has A number A name (a noun) A description One or more input data flows Usually one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes		
Every <i>external entity</i> has A name (a noun) A description	Label (name) Type (entity) Description Alias (another name) Entity description Notes		

4. Test Cases

Programming: 50 points

1. Makefile

- Macros
- Clean target

2. Programming