

CS302 QUIZ

Tuesday Dec 3, 2019

Name: ANSWER KEY

Notes, books, and electronic devices not allowed.

Problem 1 [2pts]: Inheritance (concepts)

Briefly describe the difference between program design based on composition vs inheritance.

*composition: has-a relationship
use one class within another*

*inheritance: is-a relationship
derive class from base class(es)*

Problem 2 [6pts]: Inheritance (C++ code)

```
class computer {  
    public:  
        computer();  
        virtual ~computer();  
  
        virtual void poweron()=0;  
  
    protected:  
        string OS;  
};
```

```
class laptop : public computer {  
    public:  
        laptop();  
        ~laptop();  
  
        void poweron();  
  
    private:  
        bool battery_charged();  
};
```

(a [1]) What does polymorphism mean and how does it relate to the above code?

Same named function, different behavior. Virtual base class function may have different derived class implementation.

(b [1]) What does the virtual ... =0 declaration of the computer::poweron function signify?

Function not declared within scope of class making instantiation of such object impossible. Derived classes provided implementation.

(c [1]) Why is it important that the computer destructor is declared to be virtual?

To ensure derived class destructor be called when deleting objects thru base class object pointers

(d [1]) Who has access to the protected computer::OS data member?

*Computer and laptop member functions (based and derived)
due to : public*

(e [1]) Who has access to the private laptop::battery_charged() function?

Laptop member functions only

(f [1]) When might you use generic computer pointers as opposed to laptop pointers?

When not knowing which class will be instantiated

Problem 3 [4pts]: Software Engineering (concepts)

Briefly describe what the following software development steps refer to:

- (a [1]) Requirements & Specifications: Description of WHAT software should do
(expressed in manner that can be tested)
- (b [1]) Software design: Breakdown of software into modules (classes, functions)
and their interactions (data interface etc)
- (c [1]) Software refactoring: Making software easier to maintain, extend,
understand etc (better / cleaner)
- (d [1]) Software testing: Making sure software operates within parameters
of specs (does what's req'd, doesn't fail)

Problem 4 [2pts]: Design Patterns (software design)

Briefly describe what design patterns are and why use thereof leads to better software.

Recipes for solving commonly occurring problems in software design. Leads to structured programs that are more easily maintained / modified / extended.

Problem 5 [6pts]: Computability (concepts)

- (a [1]) Which time complexity characterizes tractable problems?

Polynomial

- (b [1]) Which time complexity characterizes intractable problems?

Exponential

- (c [1]) What is a Turing machine?

Mathematical model of abstract computer

- (d [1]) What is a Turing machine used for?

Establish theoretical insights into computability / decidability

- (e [1]) State the definition of class P problems.

Problems SOLVED in polynomial time by deterministic TM

- (f [1]) State the definition of class NP problems.

Problems DECIDABLE in polynomial time by non-deterministic TM

TM \equiv Turing machine