Colin Graves COMP 333 Test#2 Review

1. a) Supply the missing code for the following program:

class Pet

{

public:

Pet(string);

void show() const; postpone binding private:

string name;

};

Pet::Pet(string newName)

{

name=newName;

}

void Pet::show() const

{

cout << " Name: "<<name;

}

class Cat Cat class inherits the public member of Pet class

{

public:

Cat( ); receives values for all data void show() const;

add a destructor

private:

string \*breed; int age;

char sex;

};

Cat::Cat( )add nec. code

{

create memory for breed on the heap

ensure memory was allocated

age=initAge validate to ensure 0-150

sex=initSex validate to ensure F or M

}

void Cat::show() const

{

cout << " Breed: " << \*breed << " Age: "<<age << " Sex: "<<sex;

display the name of the pet

}

write the destructor

void showPet( )client function that receives an Pet or Cat object

{

call the member show function

}

* 1. Declare an object of type Cat on the stack called *kitty*, passing it hard coded values.
  2. Declare an object dynamically of type Cat, passing it hard coded values. Declare any variables that you require. Ensure memory was allocated.
  3. Print the contents of the object created in b) above, by calling the member function show().
  4. Call the showPet function to print the contents of *kitty.*
  5. The function showPet() is referred to as a function.
  6. The member function show() uses binding.
  7. The object created in b) above creates memory on the .
  8. What is the difference between static and dynamic binding?