EECS 280 Lab 04 Worksheet: C++ Style ADTs

*Due Sunday, May 21, 8pm*

1. Why do we prefer to pass Classes by reference in function calls? If we do not want to modify them why do we pass them as const reference and not value?

|  |
| --- |
| Classes work with large amounts of data, thus taking up memory. Passing by reference does not make a costly copy. Passing by value will make an unnecessary copy, const reference bypasses this. |

class Dog { // The following code is relevant to

string name; // questions 2-5

int age;

public:

Dog(string name\_);

void setName(string);

string getName() const;

void have\_birthday();

};

1. Define the custom constructor for dog that initializes the name with the string name\_ and age to 0 with an initializer list

|  |
| --- |
| Dog::Dog( string name\_) : name(name\_), age(0){} |

1. Write/Define the setName and getName functions

|  |
| --- |
| void Dog::setName(string) : name( string ) {}  string Dog::getName() const{  return name;  } |

1. Write/Define the have\_birthday function to increase age

|  |
| --- |
| void Dog::have\_birthday(){  ++age;  } |

1. Which part of the dog code above would go in a .h file and which would go in a .cpp? why? (Also why did we type Dog:: in our function definitions?)

|  |
| --- |
| The function definitions 2, 3, 4 would go in a cpp file, and the declaration would go in the h. This is because if another person wanted to look at your code, they don't care how it works, they just need to know what it does.  Dog:: specifies that the function you create pertains to the Dog class. |