Practice Questions and Exercises

1. What concept of object orientated programming are we implementing when we make data members private and use member functions to access and modify them?

Encapsulation

1. True or false – A classes member functions do not have access to the classes private data members.

False

1. What does it mean to put an object into a safe and empty state?

To put an object in a safe and empty state means that an object, upon instantiation/creation, all of its data members are in a well defined, valid state.

Pointers to nullptr, integers to 0 or -1, etc

1. What are the accessibility levels in a class?

Private

Public

1. True or false – You can have private member function.

True

1. Create a class definition (what you would put in a header file) named Device with the following:
   1. private data members of the following;
      1. an integer named “\_id”
      2. an integer named “\_bitrate”
      3. a pointer to a Device named “\_child\_device”
      4. a pointer to a char named “\_name”
   2. Public member functions with the following specifications;
      1. Default constructor.;
      2. A getter/query function as well as a setter/modifier function for each private data member.

Class Device{

int \_id;

int \_bitrate;

Device\* \_child\_device;

char\* \_name;

public:

Device();

int getId();

int getBitrate();

Device\* getChild\_device();

char\* getName();

void setId(int );

void setBitrate(int );

void setChild\_device(Device\*);

void setName(char\*);

};

1. For the class created in Question 6, write the function definitions for the member functions. (What you would put in the cpp file for the class). Do not worry about validation, I want to see that you know how to write the definitions for the member functions.

int Device::getId(){

return \_id;

}

int Device::getBitrate(){

return \_bitrate;

}

Device\* Device::getChild\_device(){

return \_child\_device;

}

char\* Device::getName(){

return \_name;

}

vDevice::oid Device::setId(int id){

\_id = id;

}

void Device::setBitrate(int bit){

\_bitrate = bit;

}

void Device::setChild\_device(Device\* d ){

\_child\_device = d;

}

void Device::setName(char\* n){

strcpy(\_name, n);

}