Week 3 Exercises Submit

Sandra Batista

1.1–1.2

1. Exercise: grocerylist.cpp

1. Write a **copy constructor** for GroceryList
2. Write a **copy assignment operator** for Grocery List

Extra practice

1. Write an '==' operator for GroceryList
2. Write [] operator for GroceryList. Make sure to include const and non-const versions. (why
3. Write + operator for GroceryList
4. Write += operator for GroceryList

Code: https://github.com/sandraleeusc/csci104_fall2020_lecture/

```
// for main inside grocerylist.cpp
//once you have written appropriate
//functions, you can change main to this

int main() {
  GroceryList list1, list2;
  list1.addItem("apples");
  list1.addItem("bananas");
  list1.addItem("peaches");
  list1.printList();
  list2.addItem("onions");
  list2.addItem("peppers");
  list2.addItem("broccoli");
  GroceryList list3 = list1;
  cout << boolalpha << (list1 == list3) << endl;
 GroceryList list4 = list1 + list2;
  list4.printList();
  cout << list4[3] << endl;
  list4[3] = "oatmeal";
  list4.printList();
}
```

2. T/F and Multiple Choice Inheritance questions

Submit your solutions to these previous exam T/F and multiple choice questions on inheritance: https://github.com/sandraleeusc/csci104_fall2020_lecture/blob/master/Inheritance_q.pdf

3. Virtual Function Exercises

```
(a) class Packager {
public:
  virtual ~Packager() {}
  string package(string& s) {
    return material() + s + material();
  }
protected:
  virtual string material() { return "-"; }
private:
};
class APackager : public Packager {
protected:
  string material() { return "A"; }
};
class DblPackager : public APackager {
public:
  string package(string& s) {
    return material() + Packager::package(s) + material();
  }
};

What is output by main() and why?
```

```
int main()
{
  Packager *p = new Packager;
  APackager *a = new DblPackager;
  DblPackager *d = new DblPackager;
  string s1 = "123";
  cout << p->package(s1) << endl;
  cout << a->package(s1) << endl;
  cout << d->package(s1) << endl;
  delete p; delete a; delete d;
  return 0;
}
```

(b) https://bytes.usc.edu/cs104/resources/midterm-a.pdf Question 4 on page 7

4. Submit work on tracing functiontrace.cpp

The code for tracing is available here:
https://github.com/sandraleeusc/csci104_fall2020_lecture

Trace the output of functiontrace.cpp

- The output is in function_trace_output
- You need to understand what function is being called on each line and why.
- You should understand what function printed each statement. Other functions are called that do no print anything.
- You can add print statements to standard error, cerr
- To compile: g++ --std=c++17 -o test functiontrace.cpp
- To run and redirect standard error to a file:

./test 2> testing_outputfile