

EE208 Laboratory Session 11

Bryan Hennelly

23rd April 2018

1 Lab objectives

In this lab you will continue to work with classes in C++ that use some of the inheritance and polymorphism techniques discussed in recent lectures. Please quickly revise this lectures before proceeding. There are four questions in his lab but really all the work is in the first exercise; the exercises that follow just tweak this solution to help you learn more about polymorphism. You are required to submit a solution by Monday 30th April 5pm.

Learning Outcomes

Having completed this lab you will be able to use polymorphism for solving practical problems.

2 Questions:

Remember:

- **Comment your code.**
- **Use proper indentation for function and control structures.**
- **You should be setting up your projects using MSVS by now and making full use of the debugger to help you figure out problems yourself. If you are still having problems with this I can give a quick tutorial at the beginning of the lab. Make full use out of F10 and F11 to step into and step over lines of code.**

Exercise 1: Hewlart packers have heard good things about your recent work for Easons and they want to offer you a job to help them solve a problem they are having with their tax bill. They want you to develop some software to calculate their total tax bill owed to the tax man every month.

HP sell two type of sales product; (i) printers and (ii) a repair service. Both of these sales products have a unique ID to identify the transaction. They both have a date of sale as well as a price. VAT for the printer is 23% while VAT is charged at 13% for the repair service. If a tax bill is later that 28 days from the sales date the tax man demands 2% compound interest on the initial tax bill for every 28 days. Use inheritance to create the two types of sales products and create a number of objects of both types in your main function. Be sure to give all classes appropriate constructors, get and set values functions, as well as a print function to print out all the details. Demonstrate the functionality of your classes working in the main function. Write some code in your main function to calculate the total tax bill for all the sales items; you will need two for loops to do this assuming you have created two arrays; one for printer objects and one for repairs.

Exercise 2: Now add some code to your solution from exercise 1 to demonstrate polymorphism. Create a number of pointers to the parent class, and set these equal to the addresses of some of the objects you created as part of the first exercise. Once again demonstrate the full functionality of your classes (as much as you can) this time using the parent class pointers and the arrow notation.

Exercise 3: Now take your solution from exercise 2 and change it again. This time delete all lines where you created the original objects in exercise 1. Instead you are going to create these objects in the same line that you created the pointers as part of exercise 2. To do this write the line

parent_class_type *pointer_name = new child_class_type

You might have noticed by now that when using the parent class pointers and arrow notation, you are not able to access functions that were defined in the child classes and were not available to the parent class. To solve this problem add a little bit of code to your parent class definition to convert it to an abstract base class; see notes. Hint: You will need to use the word `virtual`. Now demonstrate the full functionality of your classes using the arrow notation.

Exercise 4: You are making great progress. Finally you need to create an array of parent class pointers and use these to point to your child class objects. Basically you need to begin with a line where you create whatever number of pointers you need to point to all of the objects that you are using.

```
parent_class_type *array_name[4]  
array_name[0] = new child_class_type etc. etc.
```

Finally use a single for loop to calculate the total tax bill.