/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// OOP244 Workshop 8: Virtual Functions

// File reflect.txt

// Version 1.0

// Date July. 22, 2019

// Author Nilan Patel

// Student Number: <038 908 117>

// Section SEE

// Description

//

//

//

// Revision History

///////////////////////////////////////////////////////////

// Name Date Reason

//

///////////////////////////////////////////////////////////

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. The difference between an abstract base class and a concrete class is that an abstract base class has functions declared and not defined. A concrete class is a class with functions that is not derived.

2. ChequingAccount::credit() shadows Account::credit()

ChequingAccount::debit() shadows Account::debit()

ChequingAccount::monthEnd() shadows SavingsAccount::monthend()

ChequingAccount::display() shadows SavingsAccount::display()

3. I learned how to derive classes from an interface and then access functions from derviced class through the base class.