Algorithms and Data Structures 2

Laboratory Sheet 4

Finish last week's lab sheet if you have not already done so. This week's lab sheet gives practice in creating structures, functions and parameter passing (Call by Value and Call by Reference).

1. Write a program which takes in two numbers and swaps them. The code for this program is in the week 4 programs on Brightspace called **swapValues.** Once the program is working correctly, change the parameter passing mechanism from `call by reference` to `call by value` and see what results your program produces.

2. Implement the *Count number of vowels* program from the lecture notes. Add an extra module to this program that takes the name as a parameter and displays it backwards.

3. Write a program that inputs an email address as a local variables in *main()*. Pass the email address as a parameter into a pure module called *generatePassword()* which adds 1 to each letter in the email address to create a password (which is a local variable in this module) Display this password in the module and display the email address in *main()*.

   Create a new version of *generatePassword()* called *generatePassword1()* which is written as a function that generates a password as before and now returns the number of characters in the password. Display the password in the function and display the email address and number of characters in the password in *main()*.

   Create another version of *generatePassword()* called *generatePassword2()* which is a pure module that takes the email address as a parameter and changes it into the password. Display the password in the module and the email address in *main()*.

4. Using the bank program in the week 4 set of programs (**arrayBankAccounts)** as a template, write a program to create an array of 3 bank customers. Assume this array is a **global variable.** Ask the user to input the customer details and then get your program to display the account numbers of customers who are overdrawn (i.e. balance < 0). Finally, ask the user to input an account number and if that account number exists, the customer details for that customer should be displayed.

   Use the following function prototypes in your program.

   ```
   void inputCustomers();
   bool overDrawn(struct bank_customer aCustomer);
   void displayCustomer(long accNum);
   ```