UECS2344 Software Design: Practical 1

1. Consider the C++ codes below:

1(a)

```
int main(void)
{
     int input;

     input = processInput();
     cout << "Number is "
          << number << endl;
}

int processInput()
{
     int input;

     cout << "Numerical input processor" << endl;
     cout << "Enter a number: ";
     cin >> input;

     return input;
}
```

1(b)

```
int main(void)
{
     int number;

     for (int i=0; i<4; i++)
     {
        cout << "Enter 1 or 2: ";
        cin >> number;
        if (number == 1)
            Output1();
        if (number == 2)
            Output2();
        else
            Output3();
     }
}

void Output1()
{
     cout << "Output1 is called" << endl;
}

void Output2()
{
     cout << "Output2 is called" << endl;
}

void Output3()
{
     cout << "Please enter only number 1 or 2." << endl;
     cout << "Thanks." << endl;
}
```

For this question, you are to submit:

- Outputs
- Structure Chart

2. Consider a login sub-system that accepts *userID* and *userPass* inputs at *Login* page from user for *MemberLogin* function. The inputs are then validated where the system then loop back to *Login* page if user failed to be authenticated. Else, the user will be brought to *Home* page which is the end of the login sub-system.

   For this question, you are to submit:

   - Structure Chart
   - Data Dictionary entries for data flows, particularly for the structure and array

3. Consider a banking application in which there are two kinds of accounts: *savings accounts* and *checking accounts*. All accounts have an account *number*, *name*, and *balance*, as well as operations to *deposit* into or *withdraw* from the account. However, the withdrawal operation for a checking account is different from that for a savings account; there is a service charge of 5% on the amount withdrawn for each withdrawal operation for a checking account. This service charge is deducted from the balance.

   Design and develop the banking application using the procedural paradigm and implemented with C++.

   The application is to have:
   - one *structure* to represent an account of the different types (using enumerated type to distinguish the type of account)
   - one *array* (for temporary storage) of many accounts
   - a console-based user interface that displays a menu with options as follows:
     - display all accounts
     - create an account
     - process an account given an account number to perform either withdraw or deposit operations for the account.

   For this question, you are to submit:

   - Source Code Listing
   - Structure Chart
   - Data Dictionary entries for data flows, particularly for the structure and array