

02 - String and Vector Class

Pre-Lab Exercises

Exercise 1

Using C++ string data type, write a programme that can detect a palindrome. A palindrome is a word that spells the same forwards and backwards (e.g. civic, radar, did, etc.). Your programme should first ask for a string input and then use functions available in the string header to determine whether it is a palindrome.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string theString;

    cout << "Enter a string to test if it is a palindrome." << endl;
    getline(cin, theString);

    int length = theString.length();
    //used for parsing string
    bool isNotPalindrome = false;
    //initially assume is a palindrome

    //compare first half with second half
    for (int i = 0; i < length/2; i++)
        if(theString[i] != theString[length - i - 1])
        {
            isNotPalindrome = true;
            break;
        } //end if
    //end for

    if (isNotPalindrome)
        cout << theString << " is not a palindrome." << endl;
    else
        cout << theString << " is a palindrome." << endl;

    return 0;
} //end main
```

Exercise 2

Go to cppreference.com and learn about vectors. Afterwards, write down all the valid methods of vector declaration with their correct syntax.

```
#include <iostream>
#include <vector>
using namespace std;

void printVector(const vector<int>& theVector)
{
    for (int i = 0; i < theVector.size(); i++)
        cout << theVector[i] << " ";
    cout << endl;
}

int main()
{
    //METHOD 1: Empty vector with capacity 0
    vector<int> firstVector;
    cout << "firstVector: "; printVector(firstVector);

    //METHOD 2: Initialisation list 1
    vector<int> secondVector {2, 3};
    cout << "secondVector: "; printVector(secondVector);

    //METHOD 3: Initialisation list 2
    vector<int> thirdVector = {4, 5, 6};
    cout << "thirdVector: "; printVector(thirdVector);

    //METHOD 4: Uninitialised with custom capacity
    vector<int> fourthVector(3);
    cout << "fourthVector: "; printVector(fourthVector);

    //METHOD 5: With arrays
    int firstArray[] = {10, 11, 12, 13};
    vector<int> fifthVector {firstArray,
        firstArray + sizeof(firstArray) / sizeof(firstArray[0])};
    cout << "fifthVector: "; printVector(fifthVector);

} //end main
```

```
firstVector:
secondVector: 2 3
thirdVector: 4 5 6
fourthVector: 0 0 0
fifthVector: 10 11 12 13
Program ended with exit code: 0
```