

## Lab 1 (Optional Questions)

### Data types, type conversion, scope, pointer

#### Q1. char data type

Complete the following steps inside the main function:

1. Declare a variable of type char, called “vChar1”, and initialize it to store the character B.
2. Declare another char variable, called “vChar2”, and initialize it to store the character ‘9’ (not the number 9).
3. Output the values of vChar1 and vChar2 using cout
4. Subtract 1 from the value of vChar1, and save the value still in vChar1.
5. Output the value of vChar1 now.

*What is the output for step 3 and 5? Can you explain why the output looks like this?  
What did you learn from this exercise?*

#### Q2. double data type

Complete the following steps, inside the main function after your code for Q1:

1. Declare a variable of type double, called “vFloat1”, initialize it to be the result of 22.0 divided by 7 (division in C++ is represented by the “/” operator).
2. Output the value of vFloat1.

#### Q3. sizeof operator

Complete the following steps, inside the main function after your code for Q2:

1. Output the number of bytes used to store an int variable, using sizeof.

Hints: You may use the following:

```
cout << "Size of int is " << sizeof(int) << " bytes.\n";
```

2. Output the number of bytes used to store a char variable, using sizeof.

3. Output the number of bytes used to store vFloat1, using sizeof.

#### Q4. Two's complement of a binary number

Use the following code as the main function. Int variables **d, c, b, a** store bits of a 4-bit binary number from high to low. Finish the program to get the two's complement representation of a given **negative** binary number.

```
void main()
{
    int a, b, c, d;
    cout << "Please input the absolute value of a negative
4-bit binary number (from -7 to 0) :" << endl;
    cin >> d >> c >> b >> a;
    cout << "Binary Number: -" << d << c << b << a << endl;
    // task: calculate the two's complement

    cout << "Two's Complement: " << d << c << b << a << endl;

}
```

**Note: Reverse the sign; Change the value of each bit; Add 1 to the resulting number to get the two's complement.**

Hint: Use additional variables to store the carry when adding digits. Use / and % to get the quotient and remainder of division.

#### Expected Outcomes

##### Example 1

```
Please input the absolute value of a negative 4-bit binary number
(from -7 to -0):
0 1 1 1
Binary Number: -0111
Two's Complement: 1001
```

##### Example 2

```
Please input the absolute value of a negative 4-bit binary number
(from -7 to -0):
0 1 0 0
Binary Number: -0100
Two's Complement: 1100
```