Lab01: Programming Review

Objectives:
To review basic concepts of C++ and get used to the VSCode and the C++ environment and errors generated.

1. The following program represent the old way C++ cast expressions. In order to cast a char variable "CH" to an Integer value, "int(CH)" is used. The new style of casting is "static_cast<int>(CH)".
a) Compile and run the program and observe the values printed.
b) Change all the old casting statements to the new casting format, then compile and run the program. Follow this style in casting your expressions from now on.

```
#include <iostream>
#include <cstdlib>
using namespace std;
main( )
{
      int i, iInt;
      char a, aChar;
      long g;
      float f;
      double d, dDouble;

      // Assign values
      i = 65;
      a = 'r';
      g = 241;
      f = 2.06;
      d = 122.448;
      cout << "The integer " << i << " converted to a character "
           << "is " << char(i) << "\n";
      cout << "The character " << a << " converted to an integer "
           << "is " << int(a) << "\n";
      cout << "The long integer " << g << " converted to a character "
           << "is " << char(g) << "\n";
      cout << "The float value " << f << " converted to an integer "
           << "is " << (int)f << "\n";
      cout << "The double value " << d << " converted to a character "
           << "is " << (char)d << "\n";
      cout << "\n\n";
      iInt = g + i;
      aChar = i + a;
      dDouble = d + g;
      cout << "The long integer " << g << " + " << " the integer "
           << i << " converted to an integer " << "is "
           << int(iInt) << "\n";

      cout << "The long integer " << g << " + " << " the integer "
           << i << " converted to a character " << "is "
           << char(iInt) << "\n";
      cout << "The integer " << i << " + " << " the character "
           << a << " converted to an integer " << "is "
           << int(aChar) << "\n";
      cout << "The integer " << i << " + " << " the character "
           << a << " converted to a character " << "is "
           << (char)aChar << "\n";
```

```
        cout << "The integer " << i << " * " << " the character "
             << a << " converted to an integer " << "is "
             << double(d * a) << "\n";
        cout << "The double " << d << " / " << " the character "
             << a << " converted to an integer " << "is "
             << int(d / a) << "\n";
        cout << "The double " << d << " + " << " the long integer "
             << g << " converted to an integer " << "is "
             << (int)dDouble << "\n";
        cout << "\n\n";

        return 0;
}
```

2. The following program displays the minimum in an array.

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

int main()
{
      // The members of the array
      int Numbers[ ] = {8, 25, 36, 44, 52, 60, 75, 89};
      int Minimum = Numbers[0];
      int a = 8;

      // Compare the members
      for (int i = 1; i < a; ++i) {
            if (Numbers[i] < Minimum)
                  Minimum = Numbers[i];
      }

      // Announce the result
      cout << "The lowest member value of the array is "
           << Minimum << "." << endl;

      return 0;
}
```

a) Add a function “ReadArray(int a[ ])” to read the elements of the array from the keyboard. Call ReadArray from the main function and run the program. The result should be the same.
b) Modify the program so that it returns the maximum value in the array.
c) Add a function to sort the data in ascending order
d) Add a function to display the contents of the array.

Sample Solution:

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

void ReadArray(int a[], int size)
{
    cout << "Enter " << size << " elements: " << endl;
    for (int i=0; i<size; i++)
    {
        cout << "-> ";
        cin >> a[i];
    }
}

int getMax(int a[], int size)
{
    int Maximum = a[0];
    for (int i = 1; i < size; ++i)
    {
        if (a[i] > Maximum)
            Maximum = a[i];
    }
    return Maximum;
}

void swap(int &x,int &y)
{
    int t = x;
```

```
    x = y;
    y = t;
}

void sortArray(int a[], int size)
{
    for (int i=0; i<size-1; i++)
        for (int j=i+1; j<size; j++)
        {   //big one bubbles to the end
            if (a[i] > a[j])
                swap(a[i],a[j]);
        }
}

void printArray(int a[],int size)
{
    for (int i = 0; i < size; ++i)
        cout << a[i] << " ";
    cout << endl;
}

int main()
{
    // The members of the array
    int Numbers[ ] = {8, 25, 36, 44, 52, 60, 75, 89};
    int Minimum = Numbers[0];
    int a = 8;
    // Compare the members
    for (int i = 1; i < a; ++i)
    {
        if (Numbers[i] < Minimum)
            Minimum = Numbers[i];
    }
    // Announce the result
    cout << "The lowest member value of the array is "
         << Minimum << "." << endl;
    cout << "\n\n";

    const int size = 4;
    int Numbers2[size];
    ReadArray(Numbers2, size);

    int Minimum2 = Numbers2[0];
    // Compare the member
    for (int i = 1; i < size; ++i)
    {
        if (Numbers2[i] < Minimum2)
            Minimum2 = Numbers2[i];
    }
    // Announce the result
    cout << "The Minumum2 = " << Minimum2 << endl;

    //compute the Maximum and announce the result
    int Maximum = getMax(Numbers2, size);
    cout << "The Maximum = " << Maximum << endl;

    sortArray(Numbers2,size);
    printArray(Numbers2,size);

    return 0;
}
```

3. Compile and run the following program. Then draw an equivalent flowchart.

```
#include <iostream>
using namespace std;
int main()
{
    char SittingDown;

    do
    {
        cout << "Are you sitting down now(y/n)? ";
        cin >> SittingDown;
        if( SittingDown != 'y' )
            cout << "Could you please sit down for the next exercise?";
        cout << "\n\n";
    }
    while( SittingDown != 'y' );

    cout << "Wonderful. Now we will continue today's exercise...\n";
    cout << "\n...\n\nEnd of exercise\n";

    char WantToContinue;

    cout << "\nDo you want to continue(1=Yes/0=No)? ";
    cin >> WantToContinue;

    if(WantToContinue == '1')
    {
        char LayOnBack;

        cout << "Good. For the next exercise, you should lay on your back";
        cout << "\nAre you laying on your back(1=Yes/0=No)? ";
        cin >> LayOnBack;

        if(LayOnBack == '0')
        {
            char Ready;

            do
            {
                cout << "Please lay on your back";
                cout << "\nAre you ready(1=Yes/0=No)? ";
                cin >> Ready;
            }
            while(Ready == '0');
        }
        else if(LayOnBack == '1')
            cout << "\nGreat.\nNow we will start the next exercise.";
        else
            cout << "\nWell, it looks like you are getting tired...";
    }
    else
        cout << "\nWe had enough today";

    cout << "\nWe will stop the session now\nThanks.\n";
    return 0;
}
```

4. Compile and fix the syntax errors found in the following program.

```
#include <iostream.h>

enum TEmploymentStatus { esFullTime, esPartTime, esContractor, esNS };

int main()
{
        int EmplStatus

        cout << "Employee's Contract Status: ";
        cout << "\n0 - Full Time | 1 - Part Time"
             << "\n2 - Contractor | 3 - Other"
             << "\nStatus: ";
        cin >> EmplStatus;
        cout << endl;

        switch( EmplStatus )
        {
        case esFullTime:
                cout << "Employment Status: Full Time\n";
                cout << "Employee's Benefits: Medical Insurance\n"
                     << " Sick Leave\n"
                     << " Maternal Leave\n"
                     << " Vacation Time\n"
                     << " 401K\n";
                break

        case esPartTime
                cout << "Employment Status: Part Time\n";
                cout << "Employee's Benefits: Sick Leave\n"
                     << " Maternal Leave\n";
                break;

        case esContractor:
                cout << "Employment Status: Contractor\n";
                cout << "Employee's Benefits: None\n"
                break;

        case esNS:
                cout << "Employment Status: Other\n";
                cout << "Status Not Specified\n";
                break;

        default:
                cout << "Unknown Status\n";
        }

        return 0;
}
```

5. Given the following program.

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

void odd(int a);
void even(int a);

int main()
{
    int i;
    do
    {
        cout << "Type a number: (0 to exit): ";
        cin >> i;
        odd(i);
    }
    while(i!=0);
    return 0;
}

void odd(int a)
{
    if ((a%2)!=0)
        cout << "Number is odd.\n";
    else
        even(a);
}

void even(int a)
{
    if ((a%2)==0)
        cout << "Number is even.\n";
    else
        odd(a);
}
```

a) Modify the program so that at the end it will display the SUM OF ALL ODD AND EVEN NUMBERS entered.
b) Modify the program to print out the MINIMUM ODD number entered and the MAXIMUM EVEN number.

6. Compile and run the following program.

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> g1;

    for (int i = 1; i <= 5; i++)
        g1.push_back(i);

    cout << "Output of begin and end: ";
    for (auto i = g1.begin(); i != g1.end(); ++i)
        cout << *i << " ";

    cout << "\nOutput of cbegin and cend: ";
    for (auto i = g1.cbegin(); i != g1.cend(); ++i)
        cout << *i << " ";

    cout << "\nOutput of rbegin and rend: ";
    for (auto ir = g1.rbegin(); ir != g1.rend(); ++ir)
        cout << *ir << " ";

    cout << "\nOutput of crbegin and crend : ";
    for (auto ir = g1.crbegin(); ir != g1.crend(); ++ir)
        cout << *ir << " ";

    return 0;
}
```

Sample Run:

```
Output of begin and end: 1 2 3 4 5
Output of cbegin and cend: 1 2 3 4 5
Output of rbegin and rend: 5 4 3 2 1
Output of crbegin and crend : 5 4 3 2 1
```