# CPT 111 – PRINCIPLES OF PROGRAMMING
# WEEK 2: PROGRAMMING LAB

1. Complete your Dev C++ installation and check your configuration. Make sure all of this sample programs can be compiled properly:

a)

```cpp
#include <iostream> //this is preprocessor directive
using namespace std;

int main () //this is the main function
{
    cout << "Hello World! Awesome, this is my first C++ program!\n";
    getchar(); //wait for the enter key to be pressed
    return 0;
}

```

b)

```cpp
// This program demonstrates using the getline function
#include <iostream>
using namespace std;

int main()
{
    string name;
    string city;

    cout << "Please enter your full name: ";
    getline(cin, name);
    cout << "Enter the city you live in: ";
    getline(cin, city);

    cout << "Hello, " << name << endl;
    cout << "You live in " << city << endl;
    return 0;
}
```

c)

```cpp
// This program stores the address of a variable in a pointer.
#include <iostream>
using namespace std;

int main()
{
   int x = 25;              // int variable
   int *ptr = nullptr;      // Pointer variable, can point to an int

   ptr = &x;      // Store the address of x in ptr
   cout << "The value in x is " << x << endl;
   cout << "The value pointed by ptr is " << *ptr << endl;
   return 0;
}
```

2. Using the program in **1 a)**, remove line 7. Explain what happened.
3. Explain what each line is for, for every line of code in **1 a)** and **1 b)**
4. Type the following program and explain how the program can produce the target achieve what it intended to do.

a)

```
1  // A simple C++ program
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "Programming is ";
8     cout << "great fun!";
9     return 0;
10 }
```

b)

```
1  // A well-adjusted printing program
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "The following items were top sellers" << endl;
8     cout << "during the month of June:" << endl;
9     cout << "Computer games" << endl;
10    cout << "Coffee" << endl;
11    cout << "Aspirin" << endl;
12    return 0;
13 }
```

Remove `endl` and make the program run again. What is the `endl` purpose/function?

c)

```
1  // Yet another well-adjusted printing program
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "The following items were top sellers\n";
8     cout << "during the month of June:\n";
9     cout << "Computer games\nCoffee";
10    cout << "\nAspirin\n";
11    return 0;
12 }
```

What is the difference between `endl` and `\n`. Replace `\n` with `endl`. Understand the difference between the two conventions.

d) Assigning variables

```
1  // This program has a variable.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7      int number;
8
9      number = 5;
10     cout << "The value in number is " << number << endl;
11     return 0;
12 }
```

e) How to use the variable in the middle of a text:

```
1  // This program has literals and a variable.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int apples;
8
9     apples = 20;
10    cout << "Today we sold " << apples << " bushels of apples.\n";
11    return 0;
12 }
```

f)

```
1  // This program has variables of several of the integer types.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int checking;
8     unsigned int miles;
9     long diameter;
10
11    checking = -20;
12    miles = 4276;
13    diameter = 100000;
14    cout << "We have made a long journey of " << miles;
15    cout << " miles.\n";
16    cout << "Our checking account balance is " << checking;
17    cout << "\nThe galaxy is about " << diameter;
18    cout << " light years in diameter.\n";
19    return 0;
20 }
```

g)

```
// This program works with characters.
#include <iostream>
using namespace std;

int main()
{
   char letter;

   letter = 'A';
   cout << letter << endl;
   letter = 'B';
   cout << letter << endl;
   return 0;
}
```

h)

```
// This program demonstrates the close relationship between
// characters and integers.
#include <iostream>
using namespace std;

int main()
{
   char letter;

   letter = 65;
   cout << letter << endl;
   letter = 66;
   cout << letter << endl;
   return 0;
}
```

What happened here? Why the numbers becoming alphabets? Where the alphabet comes from?

i)

```
// This program uses character literals.
#include <iostream>
using namespace std;

int main()
{
   char letter;

   letter = 'A';
   cout << letter << '\n';
   letter = 'B';
   cout << letter << '\n';
   return 0;
}
```

Why the answer is like that? So, in the end, where is ‘A’?

j)

```
// This program demonstrates the string class.
#include <iostream>
using namespace std;

int main()
{
   string movieTitle;

   movieTitle = "Wheels of Fury";
   cout << "My favorite movie is " << movieTitle << endl;
   return 0;
}
```

k)

```
// This program uses floating point data types.
#include <iostream>
using namespace std;

int main()
{
   float distance;
   double mass;

   distance = 1.495979E11;
   mass = 1.989E30;
   cout << "The Sun is " << distance << " meters away.\n";
   cout << "The Sun\'s mass is " << mass << " kilograms.\n";
   return 0;
}
```

What is the difference between the two variables?

l)

```
// This program demonstrates boolean variables.
#include <iostream>
using namespace std;

int main()
{
   bool boolValue;

   boolValue = true;
   cout << boolValue << endl;
   boolValue = false;
   cout << boolValue << endl;
   return 0;
}
```

m)

```
// This program determines the size of integers, long
// integers, and long doubles.
#include <iostream>
using namespace std;

int main()
{
   long double apple;

   cout << "The size of an integer is " << sizeof(int);
   cout << " bytes.\n";
   cout << "The size of a long integer is " << sizeof(long);
   cout << " bytes.\n";
   cout << "An apple can be eaten in " << sizeof(apple);
   cout << " bytes!\n";
   return 0;
}
```

n) The importance of sequence

```cpp
// This program can't find its variable.
#include <iostream>
using namespace std;

int main()
{
   cout << value; // ERROR! value not defined yet!

   int value = 100;
   return 0;
}
```

o) Example of calculation solution

This program calculates hourly wages, including overtime.

```cpp
// This program calculates hourly wages, including overtime.
#include <iostream>
using namespace std;

int main()
{
   double regularWages,             // To hold regular wages
          basePayRate = 18.25,      // Base pay rate
          regularHours = 40.0,      // Hours worked less overtime
          overtimeWages,            // To hold overtime wages
          overtimePayRate = 27.78, // Overtime pay rate
          overtimeHours = 10,       // Overtime hours worked
          totalWages;               // To hold total wages

   // Calculate the regular wages.
   regularWages = basePayRate * regularHours;

   // Calculate the overtime wages.
   overtimeWages = overtimePayRate * overtimeHours;

   // Calculate the total wages.
   totalWages = regularWages + overtimeWages;

   // Display the total wages.
   cout << "Wages for this week are $" << totalWages << endl;
   return 0;
}
```