LAB #03:

Lab Objective:

Expressions, type casting, coercion, formatting, random numbers.

Lab Description:

Implicit conversion (coercion)

Implicit conversions are automatically performed when a value is copied to a compatible type. For example

```
Short a= 2000;
Int b;
b=a;
```

When an operator works with two values of different data types, the lower-ranking value is promoted to the type of the higher-ranking value.
When the final value of an expression is assigned to a variable, it will be converted to the data type of that variable.

Type casting

C++ is a strong-typed language. Many conversions, specially those that imply a different interpretation of the value, require an explicit conversion, known in C++ as *type-casting*. There exist two main syntaxes for generic type-casting: *functional* and *c-like*:

```
double x = 10.3;
int y;
y = int (x);    // functional notation
y = (int) x;    // c-like cast notation
```

Expressions:

Multiplication, mode and division have higher precedence than addition and subtraction.

Associativity: left to right.

Random Function:

Cout << rand();

Library used <cstdlib>

Formatting:

Stream Manipulator	Description
`setw(n)`	Establishes a print field of *n* spaces.
`fixed`	Displays floating-point numbers in fixed point notation.
`showpoint`	Causes a decimal point and trailing zeroes to be displayed, even if there is no fractional part.
`setprecision(n)`	Sets the precision of floating-point numbers.
`left`	Causes subsequent output to be left justified.
`right`	Causes subsequent output to be right justified.

LAB:

Task 1:

Assume that the following variables are defined:

```
int age;
double pay;
char section;
```

Write a single cin statement that will read input into each of these variables.

Task 2:

Complete the following table by writing the value of each expression in the Value column according C++ language rules.

Expression	Value
28 / 4 – 2	
6 + 12 * 2 – 8	
4 + 8 * 2	
6 + 17 % 3 – 2	
2 + 22 * (9 - 7)	
(8 + 7) * 2	
(16 + 7) % 2 – 1	
12 / (10 - 6)	
(19 - 3) * (2 + 2) / 4	

Task 3:

Assume a program has the following variable definitions:

int units;

float mass;

double weight;

weight = mass * units;

Which automatic data type conversion will take place?

A. mass is demoted to an int, units remains an int, and the result of mass * units is an int.

B. units is promoted to a float, mass remains a float, and the result of mass * units is a float.

C. units is promoted to a float, mass remains a float, and the result of mass * units is a double.

Task 4:

Assume a program has the following variable definitions:

int a, b = 2; float c = 4.2;

and the following statement: a = b * c;

What value will be stored in a?

A. 8.4 B. 8 C. 0 D. None of the above

Task 5:

Assume that qty and salesReps are both integers. Use a type cast expression to rewrite the following statement so it will no longer perform integer division.

unitsEach = qty / salesReps;

Task 6:

Math Tutor: (hint rand())

Write a program that can be used as a math tutor for a young student. The program should display two random numbers to be added, such as

```
   247
 + 129
---------
```

The program should then pause while the student works on the problem. When the student is ready to check the answer, he or she can press a key and the program will display the correct solution:

```
   247
 + 129
```

376

Home Tasks:

Task 1:

Each of the following programs has some errors. Locate as many as you can.

Program-1

```
using namespace std;
void  main ()
{
        double number1, number2, sum;
        cout << "Enter a number: ";
        Cin << number1;
        cout << "Enter another number: ";
        cin << number2;
        number1 + number2 = sum;
        cout "The sum of the two numbers is " << sum
}
```

Program-2

```
#include <iostream>
using namespace std;
void main()
{
        int number1, number2;
        float quotient;
        cout << "Enter two numbers and I will divide\n";
        cout << "the first by the second for you.\n";
        cin >> number1, number2;
        quotient = float<static_cast>(number1) / number2;
        cout << quotient
```

}

Task 2:

Average of Values to get the average of a series of values, you add the values up and then divide the sum by the number of values. Write a program that stores the following values in five different variables: 28, 32, 37, 24, and 33. The program should first calculate the sum of these five variables and store the result in a separate variable named sum. Then, the program should divide the sum variable by 5 to get the average. Display the average on the screen.