|  |  |
| --- | --- |
| Edith Cowan University School of Science |  |

Module 2

Evaluating C++ Expressions

**Objectives:**

* Use C++ binary arithmetic operators
* Learn about the precedence and associativity of arithmetic operations
* Examine shortcut arithmetic operators
* Use other unary operators
* Evaluate Boolean expressions
* Perform operations on struct fields

**Activity:**

1. Assume a, b and c are integers and that a = 0, b = 1 and c = 5. Evaluate each of the following expressions. (Do not assume the answers are cumulative; evaluate each expression using the original values for a, b and c.)
   1. a + b
   2. a > b
   3. 3 + b \* c
   4. ++b
   5. b++
   6. b <= c
   7. a > 5
   8. ++a == b
   9. b != c
   10. b == c
   11. b = c
   12. b / c
   13. b % c
   14. b + c \* 4 / 3
   15. 22 / (c + 3)
2. Write a console application based on the following code:

/\* Note that the following code snippets are intentionally missing declarations and code, you are to rectify that in your implementation \*/

int a, b, c;

double x, y, z;

a = 13

b = 4

x = 3.3

y = 15.78

c = a + b

cout << “a + b is “ << c << endl;

z = x + y

cout << “x + y is “ << z << endl;

c = a / b

cout << “a / b is “ << c << endl;

c = a % b

cout << “a % b is “ << c << endl;

If everything worked as expected you should see the following in your console:

a + b is 17

x + y is 19.08

a / b is 3

a % b is 1

1. Using prefix and postfix increment and decrement operators, extend your program developed in question #1 with the code given below:

a = 2;

c = ++a;

cout << “a is “ << a << “ and c is “ << c << endl;

a = 2;

c = a++;

cout << “a is “ << a << “ and c is “ << c << endl;

If everything worked as expected you should see the following in your console:

a + b is 17

x + y is 19.08

a / b is 3

a % b is 1

a is 3 and c is 3

a is 3 and c is 2

1. Using operators with struct fields, write a program that uses the following code:

cout << “Please enter a student’s credit hours: “;

cin >> oneStudent.creditHours;

cout << “Please enter the student’s grade point average: “;

cin >> oneStudent.gradePointAverage;

cout << “The number of credit hours is “ << oneStudent.creditHours << endl;

cout << “The grade point average is “ << oneStudent.gradePointAverage << endl;

hoursRemaining = HOURS\_REQUIRED\_TO\_GRADUATE – oneStudent.creditHours;

cout << “This student needs “ << hoursRemaining << “ more credit hours to graduate” << endl;

If everything worked as expected you should see the following in your console:

Please enter a student’s credit hours: 38

Please enter the student’s grade point average: 3.4

The number of credit hours is 38

The grade point average is 3.4

This student needs 82 more credit hours to graduate

1. Extend your program to have a constant value for a passing grade average of 3.2. After the previous outputs to console also have it output whether the student is currently passing and whether they have already graduated (by determining if hoursRemaining is less than or equal to 0) using Boolean expressions (print 1 or 0 as per the lecture example).
   1. There will be no sample code provided for this, however the answers are in Blackboard. It is advised that you attempt to do this yourself first though.