Randy Neal

ECEN 489 HW2

1. True
2. False: In JAVA, primitive types will be identical across all platforms that have JAVA.
3. The repetition statements are the while, do…while, and for statements.
4. The syntax for (? :) is as follows: result = test condition ? value 1 : value 2

This basically states that if the test condition is true, result will be assigned the contents of value 1, if the test condition is false, then result will be assigned the contents of value 2.

1. If the initial value of c=7, then c += 3 will be 10.
2. Assuming a = 4, b = 5, c = 6, d = 15, e = 17, the table will be as follows;

|  |  |  |
| --- | --- | --- |
| Expression | Explanation | Assignment |
| a += 6 | a = a + 6 | Assign 10 to a |
| b -= 2 | b = b - 2 | Assign 3 to b |
| c \*= 3 | c = c \* 3 | Assign 18 to c |
| d /= 4 | d = d / 4 | Assign 3 to d |
| e %= 5 | e = e % 5 | Assign 2 to e |

1. The while statement is missing the { } and also it will execute indefinitely; it should be typed in this manner:

while (x >= 0)

{sum += x;

x-- ; }

1. The scope for the initialization expression in the for header is the for loop only, in other words the variable is only seen and useable inside the for loop.
2. A) for ( int index = 1; index <= 10; index++ );

B) for ( int index = 10; index <=1; index--);

C) for ( int index = 1; index <= 256; index \*= 2)

1. Methods declared with access modifier private can be called only by other methods of the class in which the private methods are declared. Such methods are commonly referred to as utility methods or helper methods because they’re typically used to support the operation of the class’s other methods.
2. The first part of the header is the initialization expression where the control variable is defined and initialized, the second part is the loop continuation condition which determines whether the loop should continue, and the increment statement determines how the control variable increments.
3. The field will be hidden until the block terminates execution, which is called shadowing.
4. Methods of the same name can be declared in the same class, as long as they have different sets of parameters (determined by the number, types and order of the parameters) this is called method overloading.