Core Java - Exercise - Operators and Numbers

Start Assignment

* **Due** No Due Date

* **Points** 100

* **Submitting** a website url or a file upload
* Write the following integers in binary notation. Do not use any Java functions or online conversion applications to calculate the answer as this will hinder the learning process and your understanding of the concept. However, you may check your answers using Java methods.
  + 1
  + 8
  + 33
  + 78
  + 787
  + 33,987

* Convert the following binary numbers to decimal notation. Do not use any Java functions or online conversion applications to calculate the answer as this will hinder the learning process and your understanding of the concept. However, you may check your answers using Java methods.
  + 0010
  + 1001
  + 0011 0100
  + 0111 0010
  + 0010 0001 1111
  + 0010 1100 0110 0111

* Write a program that declares an integer a variable x and assigns the value 2 to it and prints out the binary string version of the number ( *Integer.toBinaryString(x)* ). Now, use the left shift operator (<<) to shift by 1 and assign the result to x. Before printing the results, write a comment with the predicted decimal value and binary string. Now, print out x in decimal form and in binary notation.

Do the preceding exercise with the following values:

* + 9
  + 17
  + 88

* Write a program that declares a variable x and assigns 150 to it and prints out the binary string version of the number. Now use the right shift operator (>>) to shift by 2 and assign the result to x. Write a comment indicating what you anticipate the decimal and binary values to be. Now print the value of x and the binary string.

Do the preceding exercise with the following values:

* + 225
  + 1555
  + 32456

* Write a program that declares 3 int variables x, y, and z. Assign 7 to x and 17 to y. Write a comment that indicates what you predict will be the result of the bitwise & operation on x and y. Now use the bitwise & operator to derive the decimal and binary values and assign the result to z.

Now, with the preceding values, use the bitwise | operator to calculate the “or” value between x and y. As before, write a comment that indicates what you predict the values to be before printing them out.

* Write a program that declares an integer variable, assigns a number, and uses a postfix increment operator to increase the value. Print the value before and after the increment operator.

* Write a program that demonstrates at least 3 ways to increment a variable by 1 and does this multiple times. Assign a value to an integer variable, print it, increment by 1, print it again, increment by 1, and then print again.

* Write a program that declares 2 integer variables, x, and y, and then assigns 5 to x and 8 to y. Create another variable sum and assign the value of ++x added to y and print the result. Notice the value of the sum (should be 14). Now change the increment operator to postfix (x++) and re-run the program. Notice what the value of sum is. The first configuration incremented x and then calculated the sum while the second configuration calculated the sum and then incremented x.