# *EXAM FOR CHAPTER 1*

Use any high-level programming language you wish for the following programming exercises. Do not call built-in library functions that accomplish these tasks automatically. (Examples are sprintf and sscanf from the Standard C library.)

1. Write a function that receives a string containing a 16-bit binary integer. The function must return the string's integer value.

2. Write a function that receives a string containing a 32-bit hexadecimal integer. The function must return the string's integer value.

3. Write a function that receives an integer. The function must return a string containing the binary representation of the integer.

4. Write a function that receives an integer. The function must return a string containing the hexadecimal representation of the integer.

5. Write a function that adds two digit strings in base b, where 2 =< b =< 10 . Each string may contain as many as 1,000 digits. Return the sum in a string that uses the same number base. 6. Write a function that adds two hexadecimal strings, each as long as 1,000 digits. Return a hexadecimal string that represents the sum of the inputs.

6. Write a function that adds two hexadecimal strings, each as long as 1,000 digits. Return a hexadecimal string that represents the sum of the inputs.

7. Write a function that multiplies a single hexadecimal digit by a hexadecimal digit string as long as 1,000 digits. Return a hexadecimal string that represents the product.

8. Write a Java program that contains the calculation shown below. Then, use the javap –C command to disassemble your code. Add comments to each line that provide your best guess as to its purpose.

```
int Y;
int temp = Y + 4;
int X = temp * 3;
```

9. Devise a way of subtracting unsigned binary integers. Test your technique by subtracting binary 00000101 from binary 10001000, producing 10000011. Test your technique with at least two other sets of integers, in which a smaller value is always subtracted from a larger one.