Introduction to Data Structures **Fall 2018**

Written Assignment 1:

EECS 233 *Deadline: Sept 21, 2018 at 23:59*

Submission Guidelines: Please put all your solutions in a PDF file and submit electronically to canvas. You can either type them or write on papers and scan them. If you choose the latter, please make sure that your hand writing is clear and readable. Name your file as W1_YourCaseID_YourLastName.pdf

1. (i) Write a generic class definition for class LibraryBook that stores books that can be borrowed from the library. It creates LibrayBook objects that use title as identifiers (which would have type String), or ISBN (type integer). Provide onLoan method which takes the book ID - either title, or ISBN, and returns if the book is available in the library or not - expressed as a boolean - or some indication that the book is not found; addBook which takes the book ID, makes it available by default and stores it, and returns the indication of success or failure (e.g., if the library book capacity is reached); removeBook which removes the specified book from the library, and returns false if book is not in library. While the LibraryBook objects can be created to have either of the above two types of book IDs, the same object can't mix and match these ID types. In other words, when you create an object, you should be able to specify which ID type this object will be handling. (12 points)
Note: You can't allocate (using the "new" operator) an object or an array of objects of a generic type. So, in your constructor for LibraryBook class, you can't allocate the "books" array using a statement like this: AnyType[] books = new AnyType[100];

To avoid this, you could also try to use a workaround like this inside the constructor:
books = (AnyType[]) new Object[100];
but, depending on your Java version, it may generate warnings or not work at all.

If the above workaround does not work, you may have to allocate this array, as an array of concrete types - strings or integers - outside the constructor and pass this array to the constructor as an argument, so that the constructor signature becomes "LibraryBook(AnyType[] array)" and then assign "books = array" within the constructor.

(ii) Write the main method that (a) creates two LibraryBook objects, one for using title as IDs and the other for using ISBN as IDs; (b) adds one entry into each LibrayBook; (c) tries to find a book in each library book and prints out the result. Compile, run, and provide the printout. (6 points)

(iii) Write the main method that, again, creates two LibraryBook objects as above (one for each type of book ID), and then attempts to insert a book with ISBN as ID into the LibraryBook object that is supposed to use titles. Compile, run, and provide the printout. (6 points)

2. The below recursive program is given to compute the power of x to y. Will this program work? Why or why not? Consider invocation power(2, 3). Describe the behavior and the result of this program on this invocation. (12 points)

```
int pow( int x, int y) {
    if ( y == 0 )
        return 1;
    return ( x * pow(x, y + 1));
}
```

3. Re-write the code for reversing a string using iteration instead of recursion. (12 points)

```
String reverse(String str) {
   if (str.length() < 0)
      return str;
   return reverse(str.substring(1)) + str.charAt(0);
}
```

4. Re-write the code for computing the minimum of an array using recursion. (12 points)

```
int min( int [] a){
   int a_min = a[0];
   for( int i = 1; i < n; i++)
      if( a[i] < a_min)  a_min = a[i];
   return a_min;
}
```

5. Assuming that $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$, prove that $f_1(n) + f_2(n)$ is $O(max(g_1(n), g_2(n)))$. (10 points)

6. Please simplify (as much as possible) the following big-O expressions: (15 points)

(a) $O(n^{1.1} + nlogn)$
(b) $O(log_2(n^2) + ln(n) + 21)$
(c) $O(15n^2 + 17n^{1.2} + 4n)$

7. For each of the following program fragments provide the big-O bound of its worst-case running time, making the bound as tight as possible (but do not chase constants - remember that constants do not matter). Provide an explanation. (15 points)

(a)
```
boolean foo( int [] a, int val){
      for( int i = 1; i < n; i++) {
         if( a[i] == -1) {
            a[i] = val;
            return true;
         }
      }
      return false;
   }
```

(b)
```
int foo( int n){
      int sum = 0;
      for( int i = 0; i < n; i++)
         for( int j = 1; j < i * i; j++)
            if( j % i == 0)
               for( int k = 0; k < j; k++)
                  sum++;
      return sum;
   }
```

(c)
```
int foo( int n){
      int sum = 0;
      for( int i = 0; i < n; i++)
```

```
            for( int j = 1; j < n; j*=2)
                sum = sum + i * j;
        return sum;
    }
```

.