

## COP3530 – Assignment 1

### Objective

Students will be able to determine the relation between the growth rate of functions, use asymptotic notations, determine the time complexity of programs, and create array-based implementations of an abstract data type.

### Assignment Questions

1. (This exercise is a variation of Exercise 2.1 in Chapter 2 of the textbook) Order the following functions according to their growth rates from slowest to fastest:

$n^2, \sqrt{n}, n \log^2 n, 2^n, n, 37, n^2 \log n, n^3, n \log n$

2. Using the definition of Big-Oh discussed in class, to prove that  $10n = O(n^2)$  we can select \_\_\_\_.

- (a)  $c = 0, n_0 = 1$
- (b)  $c = 1, n_0 = 1$
- (c)  $c = 2, n_0 = 5$
- (d)  $c = 1, n_0 = 9$

3. (This exercise is a variation of Exercise 2.7 in Chapter 2 of the textbook) For each of the following program fragments, determine the time complexity:  $O(n)$ ,  $O(2^n)$ ,  $O(n^2)$ ,  $O(n \times i)$ ,  $O(i^2)$ ,  $O(n \times i^2)$ , or  $O(n^3)$ . Size of the input is  $n$  in each case.

(1) 

```
sum = 0;
for(i = 0; i < n; i++) sum++;
```

(2) 

```
sum = 0;
for(i = 0; i < n; i++)
    for(j = 0; j < n; j++)
        sum++;
```

(3) 

```
sum = 0;
for(i = 0; i < n; i++)
    for(j = 0; j < n*n; j++)
        sum++;
```

(4) 

```
sum = 0;
for(i = 0; i < n; i++)
    for(j = 0; j < i; j++)
        sum++;
```

4. Consider the Bag interface given below. Write and test an ArrayBag class that implements the interface.

```
public interface Bag
{
    public boolean isEmpty();    // returns true if bag is empty, false otherwise
    public void print();        // prints the content of the bag
    public void add(String s);   // adds string s to the bag
    public void remove(String s); // remove all occurrences of string s
    public int count(String s);  // returns number of occurrences of string s in the bag
}
```

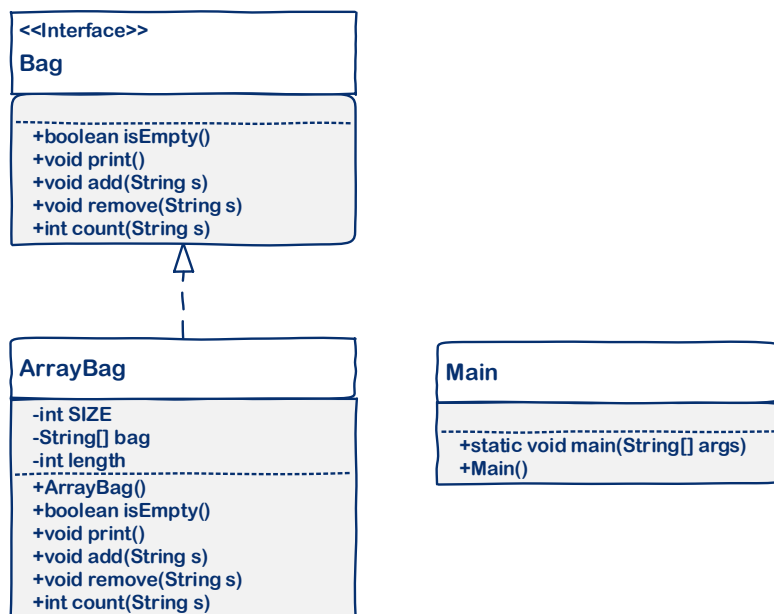
### Guidelines

The assignment is to be completed individually. Questions are based on Week 1 and 2 content.

You are allowed to use all of the code given in the lectures (for example, the implementation of the methods of the ArrayList class). In those cases, make sure you properly credit its source.

### Design

Students are expected to structure the code as indicated in the UML class diagram:



### Deliverables:

- This document only contains the instructions pertaining to the first assignment. Answers will be entered in quiz titled **Assignment 1**. Read carefully this document, answer the questions, prepare the answer to question 4 as indicated below and when you are ready, open the quiz and enter the answers.
- Questions 1, 2, and 3 will be answered on the quiz itself.

• Question 4 of the quiz is a File-Upload question. Here you will upload your answer as a compressed folder named *PID Assignment 1* (e.g. *1234567 Assignment 1*) that will contain:

- three Java files: *Bag.java*, *ArrayBag.java*, and *Main.java* (Main is the class that tests the ArrayBag class). Note: include **only** the .java files; do not include other files or folders generated by the IDE
- At least one screenshot of the running program

• Make sure you write your Panther ID as the first line of each file you submit (given as a comment in the Java files).

• Do not include your name in any of the deliverables.

### Grading Rubric

The assignment is worth 100 points (out of 1000 total course points). Grade components:

Component	Points	Description										
Submission	5	The student has submitted the project solution using the requirements for deliverables specified in the <i>Deliverables</i> section.										
Organization	5	Source code is commented, organized, and readable.										
Content	90	<table><tr><th>Question</th><th>Points</th></tr><tr><td>1</td><td>20 pts</td></tr><tr><td>2</td><td>20 pts</td></tr><tr><td>3</td><td>20 pts</td></tr><tr><td>4</td><td>30 pts</td></tr></table>	Question	Points	1	20 pts	2	20 pts	3	20 pts	4	30 pts
Question	Points											
1	20 pts											
2	20 pts											
3	20 pts											
4	30 pts											