

PURDUE UNIVERSITY

Midterm 1 Practice Examination– February 23, 2017

Problem-Solving and Object-Oriented Programming

Writing Time: TWO Hours

Question 1 (User-defined classes), 35 pts

The class `Bug` models a bug moving along a horizontal line. The bug moves either to the right or to the left. Initially, the bug moves to the right, but it can turn to change its direction. In each move, its position changes by one unit in the current direction. You will need to represent both the current position of the bug and the direction as instance variables of the class.

Within the `Bug` class:

1. Provide a constructor `public Bug(int initialPosition)`. The constructor initializes the instance variables
2. Provide the method `public void turn()`. This method changes the direction.
3. Provide the method `public void move()`. In each move the position changes by one unit.
4. Provide public "getters/accessors" for both the instance variables. Name them `getPosition` and `getDirection`.

Sample Usage:

```
Bug b = new Bug(10); // Position is 10 and direction is right
b.move(); // Now the position is 11 and direction is right
b.turn(); // Now the position is 11 and direction is left
b.move(); // Now the position is 10 and direction is left
```

Question 2 (String Manipulation), 35 pts

Complete the class `StringManipulation` by completing the method

```
public boolean haveSameChars(String s1, String s2)
```

This method checks if the two strings passed in as input arguments have the same characters. Assume that a character occurs only once in a string. In other words, every character in a string is unique. Capitalization matters. Return `true` if they have the same characters and `false` if they don't.

Special cases:

- Return `false` if either `s1` or `s2` is null
- Return `true` if `s1 = ""` and `s2 = ""`

Examples:

- Return `true` if `s1 = "Riya"` and `s2 = "Rayi"`
- Return `true` if `s1 = "123"` and `s2 = "312"`
- Return `false` if `s1 = "Riya"` and `s2 = "riya"`
- Return `false` if `s1 = "Nash"` and `s2 = "Don"`

Question 3 (Arrays), 30 pts

Write a Java class called `Large` containing a function `int[] indexOfLargest(int[] list)` to find all the indices (locations) of the "largest element" in the array. The array is passed as the argument and returns those locations as another array that contains those locations in ascending order. Examples:

- return `{0, 4}` when the input array is `{4, 0, 2, 3, 4, 0}`.
Here the largest element is 4 and it occurs at index 0 and 4.
- return `{3}` when the input array is `{0, -2, 3, 17, 0}`. Here the largest element is 17 and it occurs at index 3.

————— *End of Examination* —————