CSE 2221 - Midterm 2 Study Guide

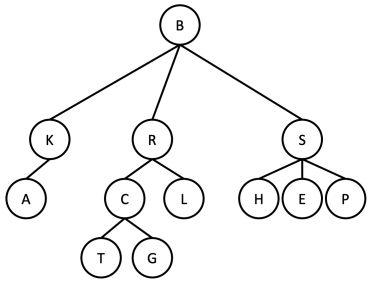
Trees

1. What is the definition of the size of a tree?

2. What is the definition of the height of a tree?

3. What is the definition of a path? What is the length of a path?

Consider the following tree for questions 5-11:



5. What is the root of the tree?

6. What are the names of the children of the root node?

7. How many children does node R have?

8. How many leaves are in the tree?

9. How many children does node E have?

10. What node is the parent of A?

11. What is the size and height of the tree?

Not given as questions, but you need to know XMLTree functions such as numberOfChildren, isTag, child, label, etc.

NaturalNumbers

13. What is the value range of a NaturalNumber? Give it in notation using square brackets and parenthesis.

14. What is the benefit of using the String constructor over the int constructor?

15. What types of methods are in the NaturalNumber interface(s)?

16. If I write “nn.isZero()”, what is “nn” called?

17. What is the name of ”nn” inside the “isZero” method?

18. What does “@ensures this = #this + 1” mean in a method contract?

19. If I write “nn.increment()”, does anything change about “nn”? What changes?

20. When should you use copyFrom over transferFrom?

21. If n1 > n2, what is the result of “n1.compareTo(n2)”?

22. In the function call “nn.multiplyBy10(k)”, what is “k” for?

23. What are the kernel methods?

24. What is the default value of a NaturalNumber?

25. What is the resulting object value of the argument passed to a transferFrom call?

References

26. What are reference types associated with?

27. Is String a primitive or reference type?

28. Is the value of a primitive type the actual value or is it a memory address?

29. Is the value of a reference type the actual value or is it a memory address?

30. How many values does a reference type have? What are they?

31. Draw the representation for the line "String myStr = "No class until March 30th lol";".

32. Draw the representation for the line "NaturalNumber myNN = new NaturalNumber2(400 + 55);"

33. What is the object value of the line of code from question 32?

34. Do we care what the reference value is? Why or why not?

35. What happens when you do "nn1 = nn2"? Assuming nn1 and nn2 are two different NaturalNumbers.

36. Is writing "a = 5, b = 5" in a tracing table the same as writing "a -> 5, b -> 5"? Assuming a and b are NaturalNumbers.

37. What gets copied in a copyFrom call?

38. What does "str2 = str1 + "ing";" do to the object value of "str1"? What about "str2"? Assuming "str1" and "str2" are two already initialized Strings. Why can we change the object value of Strings this

way, but must call methods for NaturalNumbers?

39. What is an alias? Draw the representation using NaturalNumbers. What does this look like in a tracing table?

40. What is an immutable type?

41. What is a mutable type? Don't just say the opposite of an immutable type.

42. What type is a String?

43. What type is a NaturalNumber?

44. What happens to the reference type of "str" when we execute the line "str += "hi";"? Assuming "str" is a String.

45. What happens to the reference type of "nn" when we execute the line "nn.divideBy10();"? Assuming "nn" is a NaturalNumber.

46. How are parameters passed for primitive types?

47. How are parameters passed for reference types?

48. How is "a" affected in the following method?

```
private static void budLightVirus(NaturalNumber a) {
     a = new NaturalNumber2(8);
     a.multiplyBy10();
}

public static void main(String[] args) {
     NaturalNumber a = new NaturalNumber2(24);
     budLightVirus(a);
}
```

49. What is the correct way to compare two Strings? What about NaturalNumbers?

50. What happens if you do "n1 == n2"? What about "n1.equals(n2)"? Assuming n1 and n2 are NaturalNumbers.

Arrays

51. Is an array primitive or reference?

52. What types of values can an array hold?

53. What does "=" do for arrays?

54. What does "==" do for arrays?

55. How are arrays passed to methods?

56. Draw the representation for an array of random ints (length = 5).

57. Draw the representation for an array of random NaturalNumbers (length = 4).

58. Draw the representation of the variables and trace them through the code below:

```
private static void badCoronavirusJoke(NaturalNumber n, NaturalNumber[] nns) {
     NaturalNumber temp = new NaturalNumber2(4);
     nns[0] = nns[1];
     nns[1] = nns[2];
     nns[2] = n;
     n.increment();
     n = temp;
     n.increment();
     temp.increment();
     nns[1].add(temp);
}
public static void main(String[] args) {
     NaturalNumber[] arr = new NaturalNumber2[3];
     arr[0] = new NaturalNumber2(9);
     arr[1] = new NaturalNumber2(13);
     arr[2] = new NaturalNumber2(1);
     NaturalNumber num = new NaturalNumber2(7);
     badCoronavirusJoke(num, arr);
}
```

Contracts

59. Assuming nn1 is a NaturalNumber, what value am I referring to when I say "nn1 = #nn1 + 23" in a method contract?

60. Can a reference parameter by null? If yes, do we allow it in this class?

61. What would a restores mode parameter look like in a method contract?

62. What would a clears mode parameter look like in a method contract?

63. What would a replaces mode parameter look like in a method contract?

64. What would an updates mode parameter look like in a method contract?

65. What is the default parameter mode?

66. What is the relationship between the input parameter and a replaces mode parameter?

67. When we use a parameter name in a method contract, what value are we referring to when the parameter is a (a) primitive type or (b) reference type?

68. Do we allow the following statement? "nn.multiply(nn);"

Aliasing

69. What is the simplest way to create an alias?

70. Can you ever have harmful aliasing in a no argument instance method?

71. How many "points of contact" do you have in harmless aliasing? i.e. how many reference variables reference an object.

72. How many "points of contact" do you have in harmful aliasing?

Mathematical String Notation

73. Can you have duplicate entries in a string? Does order of the entries matter?

74. Write the mathematical string notation for programming type String equal to "CSE2221". What is the mathematical type of this?

75. Write the mathematical string notation for programming type int equal to [1, 6). What is the mathematical type of this?

76. What operator do you use to concatenate two programming Strings? What about two mathematical strings?

77. What is the definition of a substring? Give an example.

78. What is the definition of a prefix? Give an example.

79. What is the definition of a suffix? Give an example.

80. What is "s[1, 7)" of "s = "Go Buckeyes!""?

Recursion Part 1

81. Simply put, what is recursion?

82. What must be done in order for recursion to work? Hint: Can I reverse "Hello" right now? No, so what must I do?

83. We keep doing recursive steps until we reach what? What is the significance of this point that we have reached?

84. What is the basic structure for a recursive method?

85. How do we trace over a recursive call? How do we determine the behavior of the call?

86. How many return statements should we have in our recursive method?

87. Considering the slow powering algorithm, what is the base case? What is the condition for a recursive step? What is our subproblem?

Object Oriented Programming

88. Fill in the blank. A Class _____ an Interface.

89. Fill in the blank. A Class _____ a Class.

90. Fill in the blank. An Interface _____ an Interface.

91. If a Class implements an Interface, how many of the methods from the Interface are implemented in the Class?

92. How can a Class call a method from its parent Class? What is the keyword?

93. If a Class extends a Class, how many of the methods from the parent Class are overrode in the child Class?

94. How do you override a method in a child Class?

95. If you have overrode a method in a child Class, can you call the original method from the parent Class?

96. What is overloading? Give an example.

97. Consider the following line: "SimpleWriter out = new SimpleWriter1L("myFile.txt");". What is the declared type? What is the object type?

98. Consider the following line: "NaturalNumber nn = new NaturalNumber2(new NaturalNumber1L(10));". What is the declared type? What is the object type?

99. What is best practice regarding the declared type?

100. What is the name of the property of Object Oriented Programming that determines which instance method (from what class) is called?

101. Can an instance method be a void return type? What are 3 examples of such methods.

Recursion Part Two (Including Recursion on Trees)

102. What do we call the approach for determining if a recursive method is correct or not?

103. What is the metric used for in a recursion?

104. If there exists a 2nd child of the root node, what can we say about this child?

105. Write a recursive function that calculates the number of edges in a tree.

106. Write a recursive function that determines how many nodes in the tree have an even number of children.

107. Write a recursive function that searches for the first occurrence of a node with n number of children.

108. What is an issue with this recursive function? Besides there being multiple returns.

```
private static int factorial(int n) {
     if (n == 3) {
      return 6;
     } else {
          return factorial(n-1) * n;
     }
}
```

109. What is an issue with this recursive function? Besides there being multiple returns.

```
private static int fibonacci(int n) {
     if (n == 1) {
      return 1;
     } else {
          return fibonacci(n-1) + fibonacci(n-2);
     }
}
```

Testing

110. What type of testing are we performing in this class?

111. What type of testing occurs when we try to combine multiple components into a system?

112. What type of testing occurs when we try to test an entire system?

113. (NOT IN SLIDES) What type of testing occurs when we try to make sure a new features has not broken old, previously working features?

114. What do we call the item being tested (using the method from #110)?

115. Draw a diagram for what it looks like for a method to be correct. Give labels.

116. Draw a diagram for what it looks like for a method to be incorrect. Give labels.

117. What are we trying to show in testing?

118. What is the difference between testing and debugging?

119. If we wrote 1000 test cases for a function and all test cases pass, is the function guaranteed to be correct?

120. What do we call a set of test cases for one method?

Consider a function "toInt()" that is overloaded to take ints, chars, doubles, and booleans and simply returns the integer representation of the input (keep in mind, chars are just 9 bit numbers deep down and booleans are just 0 or 1). The only precondition is it must be a non-negative input

121. Write 12 different good test cases for the above function (only write the function name with the input argument). Write whether they are "routine" with R, "boundary" with B, or "challenging" with C. Hint: don't have all routine test cases. You may NOT copy the ones from future questions.

122. Is toInt(-1) a good test case? Why or why not?

123. Is toInt("a") a good test case? Why or why not?

124. Is toInt(0.00000) a good test case? Why or why not?

125. Is toInt(Math.pow(2, 31)) a good test case? Why or why not?

126. Is toInt('5') a good test case? Why or why not?

127. Is toInt(new NaturalNumber2(0)) a good test case? Why or why not?

128. Write actually JUnit test cases (like ones you would write in code) for 2 of your answers from #121. What should be above your test case? What do you need in the test case to tell the program it passed?