1. Guaranteed Initialization _____ be used to initialize const data fields in a class. A) must B) can optionally C) cannot

A

2. Guaranteed Initialization _____ be used to initialize reference data fields in a class. A) must B) can optionally C) cannot

A 这个是因为是reference吧 生成的时候必须有value 不然就null了

3. Guaranteed Initialization _____ be used to initialize array data fields in a class. A) must B) can optionally C) cannot

C

4. Guaranteed Initialization _____ be used to pass parameters required to initialize object data fields in a class. A) must B) can optionally C) cannot

A

5. Guaranteed Initialization _____ be used to pass parameter required to initialize parent data fields in a class. A) must B) can optionally C) cannot

A

6. Guaranteed Initialization _____ be used to initialize primitive data fields in a class. A) must B) can optionally C) cannot

B

7. T or F: Using guaranteed initialization, data fields must be listed in the same order as they are declared in the class definition.

T (for complex scenario)

8. T or F: Using guaranteed initialization, data fields can be listed in the any order.

F

9. T or F: The semi-colon following the close curly brace in C++ at the end of a class definition is mandatory.

T

10. T or F: The semi-colon following the close curly brace in C++ at the end of a class definition is optional.

F

11. "T or F: For the following declaration, the parameter object is constant only during the method execution and not necessarily const when that method passes it to other methods: void func (const UCSDStudent &); "

F

12. "T or F: For the following declaration, the parameter object is constant both during the method execution and when that method passes it to other methods: void func (const UCSDStudent &); "

T

13. T or F: For the following declaration, the const parameter object is also a constant in the caller: void func (const UCSDStudent &);

F

14. "T or F: For the following declaration, the const parameter object is not necessarily a constant in the caller:  void func (const UCSDStudent &);"

T

15. "T or F: For the following member method declaration, any calling object cannot change during the method execution:  void func (UCSDStudent &) const;"

T

16. "T or F: For the following member method declaration, any calling object cannot change during the method execution: void func (const UCSDStudent &);"

F

17. "T or F: For the following member method declaration, any calling object cannot change during the method execution: void func (const UCSDStudent &) const;"

T

18. Using templates, what is the term for the name that is listed between < and >: Ex: SymTab ST;

Instantiation

19. T or F: When using templates with a polymorphic generic container, if the name listed between < and > is the name of an existing type, the code is likely for the user of a generic container.

T

20. T or F: When using templates with a polymorphic generic container, if the name listed between < and > is the name of an existing type, the code is likely for the implementation of a generic container.

F

21. T or F: When using templates with a polymorphic generic container, if the name listed between < and > is a name that isn't an existing type, the code is likely for the user of a generic container.

F

22. T or F: When using templates with a polymorphic generic container, if the name listed between < and > is a name that isn't an existing type, the code is likely for the implementation of a generic container.

T

23. T or F: When the parameter is a non-const reference, the method being called is expected to update the reference parameter.

T

24. "When removing from the Tree of hw8 or hw9, the _____ TNode is ""in charge"" of restructuring the pointers in the Tree. P)arent or C)hild."

Child

25. T or F: PointerInParent is another name for the pointer that was followed to get the next TNode along a search path.

T

26. In the majority of hw8 Tree code, pointerInParent was an _____ parameter. I)nput O)utput

Output

27. T or F: When using an input parameter, the caller is expecting the parameter to change.

F

28. T or F: When using an output parameter, the caller is expecting the parameter to change.

T

29. A non-const reference parameter is most likely to be an _____ parameter. I)nput O)utput

Output

30. A const reference parameter is most likely to be an _____ parameter. I)nput O)utput

Input

31. When entering a method with a pointerInParent parameter, what is the value of pointerInParent?

this

32. The successor TNode was found by going _____ once and then all the way _____. A) right, left B) left, right

A

33. The predecessor TNode was found by going _____ once and then all the way _____. A) right, left B) left, right

B

34. The successor TNode was the alphabetically _____ found in the _____ subtree. A) largest, right B) largest, left C) smallest, right D) smallest, left

C

35. The predecessor TNode was the alphabetically _____ found in the _____ subtree. A) largest, right B) largest, left C) smallest, right D) smallest, left

B

36. The _____ TNode was found by going right once and all the way left. P)redecessor S)uccessor

S

37. The _____ TNode was found by going left once and all the way right. P)redecessor S)uccessor

P

38. T or F: The successor TNode always has a null right pointer.

F

39. T or F: The successor TNode always has a null left pointer.

T

40. T or F: The predecessor TNode always has a null right pointer.

T

41. T or F: The predecessor TNode always has a null left pointer.

F

42. When finding the predecessor TNode, the code to go left once is most likely in TNode's _____ method. A) Remove B) RARM

A

43. When finding the predecessor TNode, the code to go all the way right is most likely in TNode's _____ method. A) Remove B) RARM

B

44. T or F: Before calling RARM, the data from the target TNode is saved to send back to the caller of Tree's Remove.

T

45. Assuming executing code at the predecessor TNode, write the code to overwrite the data at the targetTNode with the data of the predecessor TNode.

TargetTNode.data = data

46. T or F: The initial call to TNode's RARM is from TNode's Remove when the target TNode has 2 children.

T

47. T or F: TNode's RARM will always find a predessor TNode.

T

48. T or F: TNode's RARM will not always find a predecessor TNode.

F

49. T or F: In hw8 when exceeding the threshold on balance, the memory for the current TNode will be deallocated before SHAB returns.

T

50. T or F: In hw8 when exceeding the threshold on balance, the memory for the current TNode will be deallocated when the last line of SHAB is executed.

F

51. When exceeding the threshold on balance, at the start of SHAB where does pointerInParent point?

this

52. When exceeding the threshold on balance, after the call to TNode's Remove from SHAB where does pointerInParent point?

The deleted TNode’s replacement in the tree;

53. In hw8 besides being allocated on the RTS, what is a unique attribute of the TNode elementTNode parameter to TNode's Remove?

Not in the tree

54. In hw8 besides being a TNode not in the Tree, what is a unique attribute of the TNode elementTNode parameter to TNode's Remove?

On the RTS

55. What are the constraints on an object to be inserted into a Tree of hw8?

Provide all the methods that will be called

56. T or F: constructors for data allocated objects are called before main is called.

T

57. T or F: destructors for data allocated objects are called after main execution is completed.

T

58. T or F: If you create UCSDStudent objects with and without passing parameters to the constructor, a UCSDStudent default constructor is needed.

T

59. T or F: If you create UCSDStudent objects only by passing parameters to the constructor, a default constructor is needed.

F

60. T or F: If you create UCSDStudent objects only without passing parameters to the constructor, a UCSDStudent default constructor is needed.

F

61. Why was a Tree or parentTNode used as a parameter to the TNode constructors of hw8?

To initialize the references(data field) of the TNode

62. What is the best tree traversal to deallocate a Tree?

Post-order

63. What is the best tree traversal to display contents of a Tree to a user?

In order

64. What is the best tree traversal to make an identical copy of a Tree?

Pre-order

65. T or F: A pre-order traversal visits in the same order as the items were inserted.

F (produce the same tree. Not the same order)

66. What is the evaluation result of the following expression: cout << "Hello World\n";

cout

67. Assuming stu is a UCSDStudent, write the most equivalent code to: stu.operator<< (cout);

Stu << cout;

68. Assuming stu is a UCSDStudent, write the most equivalent code to: operator<< (stu, cout);

Stu << cout;

69. Assuming stu is a UCSDStudent, write the most equivalent code to: operator<< (cout, stu);

Cout << stu

70. Why was operator << not a member method of UCSDStudent?

Want the consistency of uses of cout

71. With a public derivation, what are the access rights on fields that originated from the public section of the parent?

public

72. With a public derivation, what are the access rights on fields that originated from the protected section of the parent?

protected

73. With a public derivation, what are the access rights on fields that originated from the private section of the parent?

Inaccessible

74. With a protected derivation, what are the access rights on fields that originated from the public section of the parent?

protected

75. With a protected derivation, what are the access rights on fields that originated from the protected section of the parent?

protected

76. With a protected derivation, what are the access rights on fields that originated from the private section of the parent?

Inaccessible

77. With a private derivation, what are the access rights on fields that originated from the public section of the parent?

private

78. With a private derivation, what are the access rights on fields that originated from the protected section of the parent?

private

79. With a private derivation, what are the access rights on fields that originated from the private section of the parent?

Inaccessible

80. Persistence means that program data exists longer than _____.

Program execution

81. What is the section of memory used to allocate all TNodes of hw9?

RTS

82. Other than the root TNode, what is the minimum number of TNodes to write to disk when inserting an item into the Tree?

2

83. "The disk file containing the Tree of hw9 is _____ file. A) an ASCII B) a binary"

B

84. What is the term for reading or writing an entire object byte by byte continuously in one operation?

serialization

85. A TNode is serialized from memory to disk by TNode's _____ method. R)ead W)rite

W

86. A TNode is serialized from disk to memory by TNode's _____ method. R)ead W)rite

R

87. What is the attribute of the TNode data that was convenient to read and write all fields as one unit of memory.

Flat data

88. When updating a hw9 TNode, what field was used to locate where it belonged on disk?

this_position

89. When inserting an item into the hw9 Tree, what field of the child TNode is assigned to the left or right field of the parent?

this_position

90. Before doing a read or write disk operation, what other disk operation is always needed?

seek

91. Translate the following hw8 code to hw9 code: left->Insert (a, left);

TNode<Whatever> leftTNode(left, fio);
leftTNode.insert(a, left);

92. Translate the following hw8 code to hw9 code: left->Remove (a, left);

TNode<Whatever> leftTNode(left, fio);
leftTNode.remove(a, left); // right syntax

93. For the ASCII file input of hw9, what is the best word that describes the program output that is ignored?

prompts

94. For the ASCII file input of hw9, besides checking for EOF, when else do we care if the input is coming from a file or from the keyboard?

Reading from file

95. For the ASCII file input of hw9, besides initiating reading from a file, when else do we care if the input is coming from a file or from the keyboard?

Checking for EOF

96. For the ASCII file input of hw9, what situation always uses cin?

The overall while loop

97. For the ASCII file input of hw9, what situation always uses cout?

results

98. T or F: For the Tree of hw9, the code still has to delete pointers to TNodes to prevent memory leaks.

F

99. T or F: For the Tree of hw9, the code no longer has to delete pointers to TNodes to prevent memory leaks.

T

100. For the Tree of hw9, how many calls need to exist in your code to TNode's deleteAllTNodes?

Zero

101. A constructor that has a reference to the same type of object being constructed is known as what kind of constructor?

Copy constructor

102. What is the name for the term for the way to sequentially access or traverse each object in a container or collection?

Iterator

103. T or F: A complete tree is one where every level (except possibly the last) is completely filled, and all nodes are as far left as possible.

T

104. T or F: A complete tree is one where every level (except possibly the last) is completely filled, and all nodes are as far right as possible.

F

105. What is the term for a tree where every level (except possible the last) is completely filled, and all nodes are as far left as possible?

complete

106. Besides maintaining "heap order", what is the other property of a binary tree that makes it a heap data structure?

Tree is complete

107. Besides inserting nodes such that the tree is complete, what is the other property of a binary tree makes it a heap data structure?

Maintaining the heap order (parent are more important than the children)

108. What is the term for the property of a heap data structure in which data of parent nodes are greater than data of child nodes?

Heap order

109. T or F: The heap order is a property of a heap data structure in which data of parent nodes are greater importance than data of child nodes.

T

110. T or F: Removing from a heap data structure is always done at the root node.

T

111. T or F: Removing from a heap data structure can be done anywhere in the tree.

F

112. What is the term for the last node that was inserted into a heap data structure?

Final leaf

113. T or F: A priority queue is often implemented via a heap data structure

T

114. T or F: A priority queue is often implemented via a binary search tree.

F

115. When using an array to implement a binary tree, what is the formula to determine the index of the parent node?

(Index - 1) / 2

116. When using an array to implement a binary tree, what is the formula to determine the index of the left child?

(Index * 2 + 1

117. When using an array to implement a binary tree, what is the formula to determine the index of the right child?

Index * 2 + 2

118. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 10?

4

119. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 9?

4

120. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 8?

3

121. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 7?

3

122. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 6?

2

123. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 5?

2

124. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 4?

1

125. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 3?

1

126. In an array-based binary tree using where the root node is at index 0, what is the index of the parent node for the node at index 2?

0

127. In an array-based binary tree using where the root node is at index 0, what is the index of the left child node for a parent node at index 1?

3

128. In an array-based binary tree using where the root node is at index 0, what is the index of the right child node for a parent node at index 1?

4

129. In an array-based binary tree using where the root node is at index 0, what is the index of the left child node for a parent node at index 2?

5

130. In an array-based binary tree using where the root node is at index 0, what is the index of the right child node for a parent node at index 2?

6

131. In an array-based binary tree using where the root node is at index 0, what is the index of the left child node for a parent node at index 3?

7

132. In an array-based binary tree using where the root node is at index 0, what is the index of the right child node for a parent node at index 3?

8

133. In an array-based binary tree using where the root node is at index 0, what is the index of the left child node for a parent node at index 4?

9

134. In an array-based binary tree using where the root node is at index 0, what is the index of the right child node for a parent node at index 4?

10

135. T or F: The sizeof a union object in C is the sizeof its largest data field.

F （T ？？）

136. What is the big-O value for the best efficiency possible?

1

Union: multiple different types of objects using the same space of memory. Change type without type cast.