## Unit 6 Test Review

<ol> <li>What does OOP stand for?</li> <li>What does ADT stand for?</li> <li>Abstract Data Type</li> <li>What does LIFO stand for?</li> <li>Last In First Out</li> <li>What does FIFO stand for?</li> <li>In First Out</li> <li>Name 6 Stack functions</li> <li>Pop, push, peek, size, isFull, isEmpty,</li> <li>Name 6 Queue functions</li> <li>Dequeue, enqueue, peek, size, isFull, isEmpty</li> <li>Name 4 functions shared by queues and stacks.</li> <li>Speed of all Stack Functions?</li> <li>Speed of all Queue Functions?</li> <li>O(1)</li> <li>Speed of binary search?</li> <li>Speed of merge?</li> <li>O(n)</li> <li>Speed of mergesort?</li> <li>O(n log n)</li> <li>Return type of toString?</li> <li>Return type of mutators?</li> <li>Return type of equals?</li> <li>Return type of equals?</li> <li>Return type of equals?</li> <li>A keyword restricting variables inside an Object's class?</li> <li>A keyword restricting variables access to inside a class</li> <li>A method that is automatically called when the variable's name is printed</li> <li>A method that checks if two objects are the same</li> <li>A method that returns 1, 0, -1</li> <li>CompareTo</li> <li>The opposite keyword to private.</li> <li>A template for an object or a type. Also contains a java programs.</li> <li>An object method type that returns the values of instance variables</li> </ol>			
3. What does LIFO stand for? 4. What does FIFO stand for? 5. Name 6 Stack functions 6. Name 6 Queue functions 7. Name 4 functions shared by queues and stacks. 8. Speed of all Stack Functions? 9. Speed of all Queue Functions? 10. Speed of binary search? 11. Speed of quicksort? 12. Speed of merge? 13. Speed of mergesort? 14. Return type of toString? 15. Return type of mutators? 16. Return type of equals? 17. Return type of equals? 18. What is the name of variables inside an Object's class? 19. A keyword restricting variables access to inside a class 20. A method that is automatically called when the variable's name is printed 21. A method that returns 1, 0, -1 23. The opposite keyword to private. 24. A template for an object or a type. Also contains a java programs. 25. An object method type that returns  15. Return type tinst is rise and that returns are the same of programs. 26. An object method type that returns Accessor	1.	What does OOP stand for?	Object Oriented Programming
4. What does FIFO stand for?  5. Name 6 Stack functions  6. Name 6 Queue functions  7. Name 4 functions shared by queues and stacks.  8. Speed of all Stack Functions?  9. Speed of all Queue Functions?  10. Speed of binary search?  11. Speed of quicksort?  12. Speed of merge?  13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is automatically called when the variable's name is printed  20. A method that returns 1, 0, -1  21. A method that returns 1, 0, -1  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  Dequeue, enqueue, peek, size, isFull, isEmpty  Pop, push, peek, size, isFull, isEmpty  peek, size, isFull,	2.	What does ADT stand for?	Abstract Data Type
5. Name 6 Stack functions Pop, push, peek, size, isFull, isEmpty, 6. Name 6 Queue functions Dequeue, enqueue, peek, size, isFull, isEmpty 7. Name 4 functions shared by queues and stacks. 8. Speed of all Stack Functions? O(1) 9. Speed of all Queue Functions? O(1) 10. Speed of binary search? O(log n) 11. Speed of quicksort? O(n log n) 12. Speed of merge? O(n) 13. Speed of mergesort? String 14. Return type of toString? String 15. Return type of mutators? Int 17. Return type of compareTo? Int 18. What is the name of variables inside an Object's class? 19. A keyword restricting variables access to inside a class 20. A method that is automatically called when the variable's name is printed 21. A method that checks if two objects are the same 22. A method that returns 1, 0, -1 CompareTo 23. The opposite keyword to private. 24. A template for an object or a type. Also contains a java programs. 25. An object method type that returns Accessor	3.	What does LIFO stand for?	Last In First Out
isEmpty,  6. Name 6 Queue functions  Dequeue, enqueue, peek, size, isFull, isEmpty  7. Name 4 functions shared by queues and stacks.  8. Speed of all Stack Functions?  9. Speed of all Queue Functions?  10. Speed of binary search?  11. Speed of quicksort?  12. Speed of merge?  13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  Dequeue, enqueue, peek, size, isFull, isEmpty  peek, size, isfulling  peek, size, isfulling  peek, size, isfulling  peek, size, isfulli	4.	What does FIFO stand for?	First In First Out
6. Name 6 Queue functions  Dequeue, enqueue, peek, size, isFull, isEmpty  7. Name 4 functions shared by queues and stacks.  8. Speed of all Stack Functions?  9. Speed of all Queue Functions?  10. Speed of binary search?  11. Speed of quicksort?  12. Speed of merge?  13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  Also contains a java programs.  25. An object method type that returns  Accessor	5.	Name 6 Stack functions	Pop, push, peek, size, isFull,
size, isFull, isEmpty  7. Name 4 functions shared by queues and stacks.  8. Speed of all Stack Functions? O(1)  9. Speed of all Queue Functions? O(1)  10. Speed of binary search? O(log n)  11. Speed of quicksort? O(n log n)  12. Speed of merge? O(n)  13. Speed of mergesort? String  14. Return type of toString? String  15. Return type of mutators? void  16. Return type of compareTo? Int  17. Return type of equals? Boolean  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1 CompareTo  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  Accessor			isEmpty,
7. Name 4 functions shared by queues and stacks.  8. Speed of all Stack Functions? O(1)  9. Speed of binary search? O(10g n)  11. Speed of quicksort? O(n log n)  12. Speed of merge? O(n log n)  13. Speed of mergesort? O(n log n)  14. Return type of toString? String  15. Return type of mutators? void  16. Return type of compareTo? Int  17. Return type of equals? Boolean  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  Accessor	6.	Name 6 Queue functions	Dequeue, enqueue, peek,
and stacks.  8. Speed of all Stack Functions? O(1)  9. Speed of all Queue Functions? O(1)  10. Speed of binary search? O(log n)  11. Speed of quicksort? O(n log n)  12. Speed of merge? O(n)  13. Speed of mergesort? O(n log n)  14. Return type of toString? String  15. Return type of mutators? void  16. Return type of compareTo? Int  17. Return type of equals? Boolean  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1 CompareTo  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns Accessor			size, isFull, isEmpty
8. Speed of all Stack Functions? O(1) 9. Speed of all Queue Functions? O(1) 10. Speed of binary search? O(log n) 11. Speed of quicksort? O(n log n) 12. Speed of merge? O(n) 13. Speed of mergesort? O(n log n) 14. Return type of toString? String 15. Return type of mutators? void 16. Return type of compareTo? Int 17. Return type of equals? Boolean 18. What is the name of variables inside an Object's class? 19. A keyword restricting variables access to inside a class 20. A method that is automatically called when the variable's name is printed 21. A method that checks if two objects are the same 22. A method that returns 1, 0, -1 CompareTo 23. The opposite keyword to private. 24. A template for an object or a type. Also contains a java programs. 25. An object method type that returns Accessor	7.	Name 4 functions shared by queues	peek, size, isFull, isEmpty
9. Speed of all Queue Functions? O(1) 10. Speed of binary search? O(log n) 11. Speed of quicksort? O(n log n) 12. Speed of merge? O(n) 13. Speed of mergesort? O(n log n) 14. Return type of toString? String 15. Return type of mutators? void 16. Return type of compareTo? Int 17. Return type of equals? Boolean 18. What is the name of variables inside an Object's class? 19. A keyword restricting variables access to inside a class 20. A method that is automatically called when the variable's name is printed 21. A method that checks if two objects are the same 22. A method that returns 1, 0, -1 CompareTo 23. The opposite keyword to private. Public 24. A template for an object or a type. Also contains a java programs. 25. An object method type that returns Accessor		and stacks.	
10. Speed of binary search?  11. Speed of quicksort?  12. Speed of merge?  13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  O(n log n)  O(n l	8.	Speed of all Stack Functions?	O(1)
11. Speed of quicksort?  12. Speed of merge?  13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  O(n log n)  O(n)  O(n)	9.	Speed of all Queue Functions?	O(1)
12. Speed of merge?  13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  O(n) log n)  O(n log n)	10.	Speed of binary search?	O(log n)
13. Speed of mergesort?  14. Return type of toString?  15. Return type of mutators?  16. Return type of compareTo?  17. Return type of equals?  18. What is the name of variables inside an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  O(n log n)  String  O(n log n)  String  Coid  Companies  Void  Instance variables  Private  ToString  equals  equals  compareTo  CompareTo  Class  Also contains a java programs.	11.	Speed of quicksort?	O(n log n)
<ul> <li>14. Return type of toString?</li> <li>15. Return type of mutators?</li> <li>16. Return type of compareTo?</li> <li>17. Return type of equals?</li> <li>18. What is the name of variables inside an Object's class?</li> <li>19. A keyword restricting variables access to inside a class</li> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1</li> <li>23. The opposite keyword to private.</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns</li> <li>26. Accessor</li> </ul>	12.	Speed of merge?	O(n)
<ul> <li>15. Return type of mutators? void</li> <li>16. Return type of compareTo? Int</li> <li>17. Return type of equals? Boolean</li> <li>18. What is the name of variables inside an Object's class?</li> <li>19. A keyword restricting variables access to inside a class</li> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private. Public</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns</li> </ul>	13.	Speed of mergesort?	O(n log n)
<ul> <li>16. Return type of compareTo? Int</li> <li>17. Return type of equals? Boolean</li> <li>18. What is the name of variables inside an Object's class?</li> <li>19. A keyword restricting variables access to inside a class</li> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private.</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	14.	Return type of toString?	String
<ul> <li>17. Return type of equals?</li> <li>18. What is the name of variables inside an Object's class?</li> <li>19. A keyword restricting variables access to inside a class</li> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private.</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	15.	Return type of mutators?	void
<ul> <li>18. What is the name of variables inside an Object's class?</li> <li>19. A keyword restricting variables access to inside a class</li> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private.</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	16.	Return type of compareTo?	Int
an Object's class?  19. A keyword restricting variables access to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  26. Accessor	17.	Return type of equals?	Boolean
<ul> <li>19. A keyword restricting variables access to inside a class</li> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private.</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	18.	What is the name of variables inside	Instance variables
to inside a class  20. A method that is automatically called when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  26. Accessor		an Object's class?	
<ul> <li>20. A method that is automatically called when the variable's name is printed</li> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private. Public</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	19.	A keyword restricting variables access	Private
when the variable's name is printed  21. A method that checks if two objects are the same  22. A method that returns 1, 0, -1 CompareTo  23. The opposite keyword to private. Public  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  Accessor		to inside a class	
<ul> <li>21. A method that checks if two objects are the same</li> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private. Public</li> <li>24. A template for an object or a type. Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	20.	A method that is automatically called	ToString
are the same  22. A method that returns 1, 0, -1  23. The opposite keyword to private.  24. A template for an object or a type. Also contains a java programs.  25. An object method type that returns  Accessor		when the variable's name is printed	
<ul> <li>22. A method that returns 1, 0, -1 CompareTo</li> <li>23. The opposite keyword to private. Public</li> <li>24. A template for an object or a type. Class Also contains a java programs.</li> <li>25. An object method type that returns Accessor</li> </ul>	21.	A method that checks if two objects	equals
<ul> <li>23. The opposite keyword to private.</li> <li>24. A template for an object or a type.</li></ul>		are the same	
<ul> <li>24. A template for an object or a type.     Also contains a java programs.</li> <li>25. An object method type that returns     Accessor</li> </ul>	22.	A method that returns 1, 0, -1	CompareTo
Also contains a java programs.  25. An object method type that returns  Accessor	23.	The opposite keyword to private.	Public
25. An object method type that returns Accessor	24.	A template for an object or a type.	Class
		Also contains a java programs.	
the values of instance variables	25.	An object method type that returns	Accessor
		the values of instance variables	

26.	An object method type that sets up	Constructor
	dynamic memory.	
27.		Facilitator
	that don't easily fit the other groups.	
28.	An object method type that sets aside	Constructor
	RAM for the instance variables and	
	initializes them.	
29.	An object method that converts the	toString
	object to a String and returns it.	
30.	An object method type that changes	Mutator
	the values of instance variables	
31.	A non-primitive type	Object
32.	Data and methods that act upon data.	Object
33.	Programmers can think of a problem	Abstraction
	at a higher more removed level.	
34.	Keeping an object's code self-	Encapsulation
	contained and independent of other	
	code.	
35.	Removing direct access to instance	Information Hiding
	variables from other programmers.	
36.	Keeps the code stable.	Information Hiding
37.	Keeps the code moveable.	Encapsulation
38.	Keeps the code useable.	Abstraction
39.	Signs of Information Hiding	1 Private Instance variables
		2 Mutators to change
		instance variables
		3 Access to view instance
		variables
40.	Signs of Abstraction	1 Easily represented by a
		physical model or diagram
		2 Code can be used without
		reading more than signatures
41.	Signs of Encapsulation	1 Easy to transfer and share
		code
		2 Easy to divide pieces among
		team

3 Class name on class line  43. Three strange things about a constructor  44. An ADT used to code an undo button  45. An ADT used for Code a line of people at a grocery store  46. An ADT used for FIFO  47. An ADT used for LIFO  48. An ADT used to store recursive method calls  49. An ADT used to store a back button.  50. An ADT used to model cars on a road  51. An ADT used to model loading an airplane.  52. An ADT to hold print jobs for a printer  53. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front.  56. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and  51. Stack  52. An ADT where you add to the back and remove from the back.  55. An ADT where you add to the back and remove from the back.	42.	Three uses of class name	1 File name to save as 2 Constructor name
43. Three strange things about a constructor  1 No return type on method signature, it returns itself 2 Same name as class 3 Called using new  44. An ADT used to code an undo button 45. An ADT used to code a line of people at a grocery store  46. An ADT used for FIFO 47. An ADT used for LIFO 48. An ADT used to store recursive method calls 49. An ADT used to store a back button. 50. An ADT used to model cars on a road 51. An ADT used to model loading an airplane. 52. An ADT to hold print jobs for a printer 53. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well. 55. An ADT where you add to the front and remove from the front 56. An ADT where you add to the back and remove from the back 58. An ADT where you add to the back and remove from the back 59. An ADT where you add to the top and 51. Stack 53. An ADT where you add to the back and remove from the back 54. An ADT where you add to the back and remove from the back 55. An ADT where you add to the back and remove from the back 56. An ADT where you add to the back and remove from the back 57. An ADT where you add to the back and remove from the back 58. An ADT where you add to the top and 59. An ADT where you add to the top and 50. Stack 51. An ADT where you add to the back and remove from the back 59. An ADT where you add to the top and			
2 Same name as class 3 Called using new  44. An ADT used to code an undo button  45. An ADT used to code a line of people at a grocery store  46. An ADT used for FIFO  47. An ADT used for LIFO  48. An ADT used to store recursive method calls  49. An ADT used to store a back button.  50. An ADT used to model cars on a road 51. An ADT used to model loading an airplane.  52. An ADT to hold print jobs for a printer  53. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front  56. An ADT where you add to the back and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and  5tack	43.		1 No return type on method
3 Called using new  44. An ADT used to code an undo button  45. An ADT used to code a line of people at a grocery store  46. An ADT used for FIFO  47. An ADT used for LIFO  48. An ADT used to store recursive method calls  49. An ADT used to model cars on a road  50. An ADT used to model loading an airplane.  51. An ADT to hold print jobs for a printer  52. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back  59. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and  5tack		constructor	
<ul> <li>44. An ADT used to code an undo button</li> <li>45. An ADT used to code a line of people at a grocery store</li> <li>46. An ADT used for FIFO</li> <li>47. An ADT used for LIFO</li> <li>48. An ADT used to store recursive method calls</li> <li>49. An ADT used to model cars on a road</li> <li>50. An ADT used to model loading an airplane.</li> <li>51. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front.</li> <li>56. An ADT where you add to the front and remove them the front.</li> <li>57. An ADT where you add to the back and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> </ul>			
<ul> <li>45. An ADT used to code a line of people at a grocery store</li> <li>46. An ADT used for FIFO Queue</li> <li>47. An ADT used for LIFO Stack</li> <li>48. An ADT used to store recursive method calls</li> <li>49. An ADT used to model cars on a road Queue</li> <li>50. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer Queue</li> <li>53. An ADT to hold changes to a bank Queue account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front.</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. Stack</li> <li>54. Stack</li> <li>55. An ADT where you add to the back and remove from the back</li> <li>56. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>54. Stack</li> </ul>			
at a grocery store  46. An ADT used for FIFO  47. An ADT used for LIFO  48. An ADT used to store recursive method calls  49. An ADT used to store a back button.  50. An ADT used to model cars on a road Queue  51. An ADT used to model loading an airplane.  52. An ADT to hold print jobs for a printer Queue  53. An ADT to hold changes to a bank queue account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front.  56. An ADT where you add to the front queue queue and remove them the front.  57. An ADT where you add to the front queue queue and remove from the back  58. An ADT where you add to the back and queue q	-		
<ul> <li>46. An ADT used for FIFO</li> <li>47. An ADT used for LIFO</li> <li>48. An ADT used to store recursive method calls</li> <li>49. An ADT used to model cars on a road</li> <li>50. An ADT used to model loading an airplane.</li> <li>51. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. An ADT where you add to the back and remove from the back</li> <li>54. An ADT where you add to the front and remove from the back</li> <li>55. An ADT where you add to the front and remove from the back</li> <li>56. An ADT where you add to the back and remove from the back</li> <li>58. An ADT where you add to the top and</li> <li>59. An ADT where you add to the top and</li> </ul>	45.	·	Queue
<ul> <li>47. An ADT used for LIFO</li> <li>48. An ADT used to store recursive method calls</li> <li>49. An ADT used to store a back button.</li> <li>50. An ADT used to model cars on a road Queue</li> <li>51. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer Queue</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> </ul>			
48. An ADT used to store recursive method calls  49. An ADT used to store a back button. Stack  50. An ADT used to model cars on a road Queue  51. An ADT used to model loading an airplane.  52. An ADT to hold print jobs for a printer Queue  53. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front  56. An ADT where you add to the back and remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack	46.	An ADT used for FIFO	Queue
method calls  49. An ADT used to store a back button. Stack  50. An ADT used to model cars on a road Queue  51. An ADT used to model loading an airplane.  52. An ADT to hold print jobs for a printer Queue  53. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front  56. An ADT where you add to the back and remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack	47.	An ADT used for LIFO	Stack
<ul> <li>49. An ADT used to store a back button.</li> <li>50. An ADT used to model cars on a road Queue</li> <li>51. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer Queue</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. Stack</li> <li>54. An ADT where you add to the front and remove from the back</li> <li>55. An ADT where you add to the back and remove from the back</li> <li>56. An ADT where you add to the back and remove from the back</li> <li>57. An ADT where you add to the top and</li> <li>58. An ADT where you add to the top and</li> <li>59. An ADT where you add to the top and</li> </ul>	48.	An ADT used to store recursive	Stack
<ul> <li>50. An ADT used to model cars on a road Queue</li> <li>51. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer Queue</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. Stack</li> <li>54. Stack</li> <li>55. Stack</li> <li>56. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> </ul>		method calls	
<ul> <li>51. An ADT used to model loading an airplane.</li> <li>52. An ADT to hold print jobs for a printer Queue</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. Stack</li> <li>54. Stack</li> <li>55. Stack</li> <li>56. Stack</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> </ul>	49.	An ADT used to store a back button.	Stack
airplane.  52. An ADT to hold print jobs for a printer Queue  53. An ADT to hold changes to a bank account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front  56. An ADT where you add to the back and remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack	50.	An ADT used to model cars on a road	Queue
<ul> <li>52. An ADT to hold print jobs for a printer</li> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. Stack</li> <li>54. Stack</li> <li>55. Stack</li> <li>56. An ADT where you add to the front and remove from the back</li> <li>56. Stack</li> <li>57. Stack</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>50. Stack</li> </ul>	51.	An ADT used to model loading an	Queue
<ul> <li>53. An ADT to hold changes to a bank account</li> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. Stack</li> <li>53. Stack</li> <li>54. Stack</li> <li>55. Stack</li> <li>56. Stack</li> <li>57. An ADT where you add to the front and stack</li> <li>58. An ADT where you add to the back and stack</li> <li>59. An ADT where you add to the top and</li> <li>50. Stack</li> </ul>		airplane.	
account  54. An ADT that could model a pile of books well.  55. An ADT where you add to the front and remove from the front  56. An ADT where you add to the back and remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack	52.	An ADT to hold print jobs for a printer	Queue
<ul> <li>54. An ADT that could model a pile of books well.</li> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and Stack</li> </ul>	53.	An ADT to hold changes to a bank	Queue
books well.  55. An ADT where you add to the front and remove from the front  56. An ADT where you add to the back and remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack		account	
<ul> <li>55. An ADT where you add to the front and remove from the front</li> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and</li> <li>51. Stack</li> <li>52. An ADT where you add to the top and</li> <li>53. Stack</li> <li>54. Stack</li> <li>55. An ADT where you add to the top and</li> <li>56. Stack</li> <li>57. Stack</li> <li>58. Stack</li> <li>59. An ADT where you add to the top and</li> <li>50. Stack</li> </ul>	54.	An ADT that could model a pile of	Stack
and remove from the front  56. An ADT where you add to the back and remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack		books well.	
<ul> <li>56. An ADT where you add to the back and remove them the front.</li> <li>57. An ADT where you add to the front and remove from the back</li> <li>58. An ADT where you add to the back and remove from the back.</li> <li>59. An ADT where you add to the top and Stack</li> </ul>	55.	An ADT where you add to the front	Stack
remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack		and remove from the front	
remove them the front.  57. An ADT where you add to the front and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack	56.	An ADT where you add to the back and	Queue
and remove from the back  58. An ADT where you add to the back and remove from the back.  59. An ADT where you add to the top and Stack		-	
<ul><li>58. An ADT where you add to the back and remove from the back.</li><li>59. An ADT where you add to the top and Stack</li></ul>	57.	An ADT where you add to the front	Queue
remove from the back.  59. An ADT where you add to the top and Stack		and remove from the back	
remove from the back.  59. An ADT where you add to the top and Stack	58.	An ADT where you add to the back and	Stack
, ' '		•	
	59.	An ADT where you add to the top and	Stack
ı		remove from the top.	

60.	Stack tradeoff	Positive: All functions O(1), very fast Negative: Only LIFO functions. Speed is achieved through
61.	Queue tradeoff	Iimiting functionality Positive: All functions O(1), very fast Negative: Only FIFO functions. Speed is achieved through Iimiting functionality
62.	How do you switch a Queue to a Queue of a Frog class?	Switch all Objects to Frog
63.		Stack s = new Stack(); Stack s2 = new Stack();
64.	Write code that declares a queue	Queue q = new Queue();
65.	Write code that prints out a stack	while(! s.isEmpty()) System.out.println(s.pop());
66.	Write code that switches the top two elements of a stack	<pre>Int holder = s.pop(); Int holder2 = s.pop(); s.push(holder); s.push(holder2);</pre>
67.	Write code that empties (and reverses) one stack into another	while(!.s.isEmpty()) S2.push(s.pop());
68.	Write code that empties a queue (and reverses) a queue into a stack	while (!q.isEmpty()) s.push(q.dequeue());
69.	How do you see the top element of a stack?	System.out.println(s.peek());
70.	What is half the size of a stack?	s.size/2