632876
We say indexing is fast if it's done in O(1) time, searching is fast if done in O(lg N) time, and inserting and deleting are fast if done in O(1) time. Compared to other data structures, unsorted arrays offer:

a. slow indexing, slow search, slow insertions and deletions.

b. fast indexing, fast search, slow insertions and deletions.

*c. fast indexing, slow search, slow insertions and deletions.

d. slow indexing, slow search, fast insertions and deletions.

e. slow indexing, fast search, fast insertions and deletions.

f. "
g. "
h. "
i. "
j. "

General Feedback:

Unsorted arrays can be indexed in constant time (which is fast), searched in O(N) time (which is not as good as O(lg N)), and restructured in O(N) time (which is not as good as O(1) time).

634933
One of the things in Java that allows us to use polymorphism is that the declared type and actual type of a variable may be different.

In Java, the actual type of a parameter or variable’s value can be any concrete class that is

a. the same as the declared type, or any subclass of the declared type (if the declared type is a class)

b. any class that implements the declared type (if the declared type is an interface)

c. any subclass of a class that implements the declared type (if the declared type is an interface)

*d. A, B, and C above.

e. A and B above, but not C

f. "
g. "
h. "
i. "
j. "

General Feedback:

The rule of thumb is that the declared type is the same or more abstract than the actual type. If the declared type is a class, the things that are equally or less abstract are its descendants -- a subclass specializes a class. If declared type is an interface, it is an abstraction of any class that implements it, or by the relationship between classes and its descendants, any class that is a subclass of something that implements it.

Java wants a guarantee that the actual type will have the declared type's methods.  That can be via inheritance or implementing an interface, or a combination of the two.

632218
An example of something that could be implemented using a Stack is:

a. The undo operation in Word

b. The back button in a web browser

c. A postfix calculator

*d. All of the above

e. Items (A) and (B) only

f. "
g. "
h. "
i. "
j. "

General Feedback:

Anytime you're solving a problem where you're adding and removing data and you always need to remove the item that was added most recently is suited to a Stack. This is called a last-in-first-out property, or LIFO. Back buttons, the undo operation, and postfix calculators all have the last-in-first-out property.