

Use the program description below for Questions 18–20.

A car dealer needs a program that will maintain an inventory of cars on his lot. There are three types of cars: sedans, station wagons, and SUVs. The model, year, color, and price need to be recorded for each car, plus any additional features for the different types of cars. The program must allow the dealer to

- Add a new car to the lot.
- Remove a car from the lot.
- Correct any data that's been entered.
- Display information for any car.

18. The programmer decides to have these classes: `Car`, `Inventory`, `Sedan`, `SUV`, and `StationWagon`. Which statement is *true* about the relationships between these classes and their attributes?

- I There are no inheritance relationships between these classes.
- II The `Inventory` class *has-a* list of `Car` objects.
- III The `Sedan`, `StationWagon`, and `SUV` classes are independent of each other.

- Correct any data that's been entered.
- Display information for any car.

18. The programmer decides to have these classes: Car, Inventory, Sedan, SUV, and StationWagon. Which statement is *true* about the relationships between these classes and their attributes?

- I There are no inheritance relationships between these classes.
- II The Inventory class *has-a* list of Car objects.
- III The Sedan, StationWagon, and SUV classes are independent of each other.

- (A) I only
- (B) II only
- (C) III only
- (D) I and II only
- (E) II and III only

19. Suppose that the programmer decides to have a Car class and an Inventory class. The Inventory class will maintain a list of all the cars on the lot. Here are some of the methods in the program:

19. Suppose that the programmer decides to have a Car class and an Inventory class. The Inventory class will maintain a list of all the cars on the lot. Here are some of the methods in the program:

```
addCar           //adds a car to the lot
removeCar        //removes a car from the lot
displayCar       //displays all the features of a given car
setColor         //sets the color of a car to a given color
                 //May be used to correct data
getPrice         //returns the price of a car
displayAllCars   //displays features for every car on the lot
```

In each of the following, a class and a method are given. Which is the *least* suitable choice of class to be responsible for the given method?

- (A) Car, setColor
- (B) Car, removeCar
- (C) Car, getPrice
- (D) Car, displayCar
- (E) Inventory, displayAllCars

20. Suppose Car is a superclass and Sedan, StationWagon, and SUV are subclasses of Car. Which of the following is the most likely method of the Car class to be overridden by at least one of the subclasses (Sedan, StationWagon, or SUV)?

- (A) setColor(newColor) //set color of Car to newColor
- (B) getModel() //return model of Car
- (C) displayCar() //display all features of Car
- (D) setPrice(newPrice) //set price of Car to newPrice
- (E) getYear() //return year of Car

21. What is the result of running this code segment?

```
Map<String, String> dwarfs = new HashMap<String, String>();  
dwarfs.put("Sneezy", "sick dwarf");  
dwarfs.put("Happy", "merry dwarf");  
dwarfs.put("Grumpy", "irritable dwarf");  
String s = dwarfs.get("Dopey");
```

- (C) `displayCar()` //display all features of Car
- (D) `setPrice(newPrice)` //set price of Car to newPrice
- (E) `getYear()` //return year of Car

21. What is the result of running this code segment?

```
Map<String, String> dwarfs = new HashMap<String, String>();  
dwarfs.put("Sneezy", "sick dwarf");  
dwarfs.put("Happy", "merry dwarf");  
dwarfs.put("Grumpy", "irritable dwarf");  
String s = dwarfs.get("Dopey");
```

- (A) A `NoSuchElementException` will be thrown.
- (B) An `IllegalStateException` will be thrown.
- (C) A `ClassCastException` will be thrown.
- (D) The code will run without error, and `s` will have the value "Dopey".
- (E) The code will run without error, and `s` will have the value `null`.

22. Assume that `ArrayList a` is initialized with `SomeType` elements. Also, assume the existence of the following method:

```
//Postcondition: Returns a HashSet that contains all the elements
//                of ArrayList<SomeType> list.
public HashSet<SomeType> copyListToHashSet(ArrayList<SomeType> list)
```

Consider the following code segment:

```
HashSet<SomeType> s = copyListToHashSet(a);
System.out.println("Number of elements in ArrayList is " + a.size());
System.out.println("Number of elements in HashSet is " + s.size());
```

Suppose the output produced by this code segment is

```
Number of elements in ArrayList is 10
Number of elements in HashSet is 6
```

Which is a valid conclusion?

- (A) List `a` contains ten distinct (i.e., different) elements, and set `s` contains six distinct elements.
- (B) There is at least one element in list `a` that occurs more than once.
- (C) List `a` contains four more distinct elements than set `s`.
- (D) There are at least four elements in list `a` that occur more than once.
- (E) There is one element in list `a` that occurs five times.