

✓ Which TWO of the following statements are TRUE? *

- ☐ Exception handling can be used to perform cleanup operations, such as closing open files or releasing system resources, when an exception occurs.
- ☒ Exception handling allows developers to gracefully handle errors or exceptional conditions that may occur during program execution. ✓
- ☐ Exception handling can only be used to handle runtime exceptions, not checked exceptions.
- ☒ The try-catch block is a common construct used in exception handling, where the code that may throw an exception is placed within the "try" block, and the code to handle the exception is placed within the "catch" block. ✓

- ✓ Which TWO of the following statements are FALSE about the code below? ★
(Select 2 FALSE statements)

```
public interface Vehicle {  
    void start();  
    void stop();  
}  
  
public abstract class Car implements Vehicle {  
    private String brand;  
  
    public Car(String brand) {  
        this.brand = brand;  
    }  
  
    public void start() {  
        System.out.println("Starting the " + brand + " car.");  
    }  
  
    public abstract void drive();  
  
    public void stop() {  
        System.out.println("Stopping the " + brand + " car.");  
    }  
}  
  
public class Sedan extends Car {  
    public Sedan(String brand) {  
        super(brand);  
    }  
  
    public void drive() {  
        System.out.println("Driving the " + brand + " sedan.");  
    }  
}
```

- ☐ The Sedan class provides an implementation for the drive() method.
- ☐ The Car class is an abstract class that implements the Vehicle interface.
- ☒ The Sedan class is an interface that extends the Car class. ✓
- ☐ The start() and stop() methods in the Car class override the corresponding methods from the Vehicle interface.
- ☐ The start() and stop() methods in the Vehicle interface are public abstract methods.
- ☒ The Car class provides an implementation for the drive() method. ✓
- ☐ The Car class has a constructor that takes a "brand" parameter.

✓ Which THREE of the following statements are TRUE? *

```
import java.util.Scanner;

public void readDataFromFile(String filename) {
    try {
        File file = new File(filename);
        Scanner scanner = new Scanner(file);
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            System.out.println(line);
        }
        scanner.close();
    } catch (FileNotFoundException e) {
        System.out.println("Error: File not found.");
    }
}
```

- ☐ The code writes data to a file instead of reading from a file.
- ☒ The code is missing the import statement for the java.io.File classes ✓
- ☒ The code is missing the import statement for the java.io.FileNotFoundException classes ✓
- ☐ The code automatically creates a new file if the specified file does not exist.
- ☒ Class declaration is missing. ✓

- ✓ The following method searches a target from an array named "ID". It returns ★ true if the target is found in ID, otherwise returns false.

4 lines out of the 7 lines of code below when arrange in the correct sequence, will perform the task above.

Fill in the blanks with 4 line numbers in the correct sequence. For example, if the order/sequence is line 3, line 2, line 1 and line 4, then you should write:
3214

(*NO SPACE IN YOUR ANSWER, only 4 line numbers is required, extra space written in answer will be given zero)

```
public static boolean search (String[] ID, String target){  
  
    //Select 4 lines that should be written here from the 7 lines below.  
  
}  
  
1: if(target.compareTo(ID[i]))  
2: return true;  
3  if (target == (ID[i]))  
4: for (int i = 0; i < ID.length; i++)  
5: return false;  
6: if (target.equals(ID[i]))  
7: for (int i = 0; i < ID.length(); i++)
```

4625



✓ Which two of the following statements about stacks, queues, and lists are FALSE? *

- ☐ Stacks and queues are examples of linear data structures.
- ☐ A linked list is a dynamic data structure that can efficiently insert and delete elements at both ends.
- ☒ A stack follows the FIFO (First-In, First-Out) principle. ✓
- ☒ A priority queue orders elements based on their insertion order. ✓
- ☐ An ArrayList is an example of a list implementation in Java.
- ☐ A queue follows the FIFO (First-In, First-Out) principle.

- ✓ The following method searches a target from an array named "ID". It returns ★ true if the target is found in ID, otherwise returns false.

4 lines out of the 7 lines of code below when arrange in the correct sequence, will perform the task above.

Fill in the blanks with 4 line numbers in the correct sequence. For example, if the order/sequence is line 3, line 2, line 1 and line 4, then you should write:
3214

(*NO SPACE IN YOUR ANSWER, only 4 line numbers is required, extra space written in answer will be given zero)

```
public static boolean search (String[ ] ID, String target){  
  
    //Select 4 lines that should be written here from the 7 lines below.  
  
}  
  
1: if(target.compareTo(ID[i]))  
2: return true;  
3  if (target == (ID[i]))  
4: for (int i = 0; i < ID.length; i++)  
5: return false;  
6: if (target.equals(ID[i]))  
7: for (int i = 0; i < ID.length(); i++)
```

4625

