

1

(There are multiple drop-downs below. Please select a correct response for each drop-down)

```
import java.util.Random;
public class User {

    String name;
    String password;

    public User(String name, String password) {
        this.name = name;
        this.password = password;
    }
    public static void main(String args[]) {
        User user = new User("Jack", generatePassword());
        System.out.println("User is "+user.name+ " and password is "+user.password);
    }
    private static String generatePassword() {
        String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
        Integer characterLength = characters.length();

        String password = "";
        Random random = new Random();
        for(int i = 0; i <= 5; i++) {
            System.out.println(password);

            password += characters.charAt(random.nextInt(characterLength));
        }
        return password;
    }
}
```

2

Topic: Java

The function `cleanTxtFiles()` given below, removes all non-alphabet characters from all the ".txt" files in a specific folder. Review the code and complete the missing parts.

(There are multiple drop-downs in the below code, please select a correct response for each drop-down)

```
void cleanTxtFiles {
    Path path = Paths.get("/Users/arezk/Documents/test/");
    String regex = "[^A-Z a-z]";

    Files.walkFileTree(path, new SimpleFileVisitor<Path>() {
        public FileVisitResult visitFile(Path file, BasicFileAttributes attrs) throws IOException {
            if (file.getFileName().toString().endsWith(".txt")) {
                try (Stream<String> lines = Files.lines(file)) {
                    BufferedWriter bufferedWriter = new BufferedWriter(file.toAbsolutePath() + ".clean");
                    lines.map(s -> s.replaceAll(regex, "")).forEachOrdered(output::flush);
                }
                return FileVisitResult.CONTINUE;
            }
        });
    });
}
```

3

Topic: Kafka

In Kafka, what is the logical feed name to which records are published and supports a multi-subscriber model so that a topic can have zero, one, or many consumers that subscribe to the data written to it?

A Epic

B Story

C Topic

D Feed

4

```
class Consumer {  
    public static void main(String[] args) {  
        Properties properties = new Properties();  
        properties.put("bootstrap.servers", "localhost:9092");  
        properties.put("key.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");  
        properties.put("value.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");  
        properties.put("group.id", "test-group");  
  
        KafkaConsumer kafkaConsumer = new KafkaConsumer(properties);  
        List topics = new ArrayList();  
        topics.add("topic-test");  
  
        kafkaConsumer.subscribe(topics);  
  
        try{  
            while (true){  
                ConsumerRecords records = kafkaConsumer.poll(10);  
  
                Iterator a = records.iterator();  
                while (a.hasNext() ){  
  
                    ConsumerRecord b = (ConsumerRecord) a.next();  
  
                    System.out.println(String.format("Topic - %s, Value: %s",  
                        b.topic(), b.value()));  
                }  
            }catch (Exception e){  
                System.out.println(e.getMessage());  
            }finally {  
                kafkaConsumer.close();  
            }  
        }  
    }  
}
```

5

Topic: Spring Boot

You have a student management system developed using the Spring Boot Framework. Also, when you implement Spring Security in your application, it removes all vulnerabilities. Complete the following code to implement Spring Security.

Note: you take the authentication credentials from the database, such as username and password.

(There are multiple drop-downs below. Select a correct response for each drop-down)

```
public class MySpringSecurityApplication extends [ ] {  
    @Autowired  
    DataSource source;  
}  
  
@Override  
protected void configure(AuthenticationManagerBuilder auth)  
throws Exception  
{  
    auth.[ ].dataSource(source);  
}
```

6

Topic: NoSQL

Complete the below code to update the marks in the subject "English" for the student with roll = 5.

(There are multiple drop-downs below. Please select the correct response for each drop-down)

```
import com.datastax.driver.core.Cluster;  
import com.datastax.driver.core.Session;  
  
public class Update_Data {  
    public static void main(String args[]){  
        //query  
        String query = "UPDATE results SET subject_name='English', marks=90 WHERE sroll = 5;"  
        //Creating Cluster object  
        Cluster cluster = Cluster.builder().addContactPoint [ ] ("127.0.0.1").build();  
        //Creating Session object  
        Session session = cluster.connect [ ] ("tp");  
        //Run the query  
        session.execute [ ] (query);  
        System.out.println("Data updated");  
    }  
}
```

7

```
1. @Configuration  
2.  
3. public class CarAutoConfiguration {  
4.     @Bean  
5.     public CarService carService() { ... }  
6. }
```



You have a requirement to include the configurations of *CarService* bean only if it is not present already in the application context. Which of the following is correct?

A

```
1. @Configuration  
2. @ConditionalOnClass  
3. public class CarAutoConfiguration {  
4.     @Bean  
5.     public CarService carService() { ... }  
6. }
```

B

```
1. @Configuration  
2. @ConditionalOnMissingClass  
3. public class CarAutoConfiguration {  
4.     @Bean  
5.     public CarService carService() { ... }  
6. }
```

C

```
1. @Configuration  
2. public class CarAutoConfiguration {  
3.     @Bean  
4.     @ConditionalOnBean  
5.     public CarService carService() { ... }  
6. }
```

hire.glider.ai 正在共享你的屏幕。 停止共享 隐藏

You have a requirement to include the configurations of *CarService* bean only if it is not present already in the application context. Which of the following is correct?

A

```
1. @Configuration  
2. @ConditionalOnClass  
3. public class CarAutoConfiguration {  
4.     @Bean  
5.     public CarService carService() { ... }  
6. }
```

B

```
1. @Configuration  
2. @ConditionalOnMissingClass  
3. public class CarAutoConfiguration {  
4.     @Bean  
5.     public CarService carService() { ... }  
6. }
```

C

```
1. @Configuration  
2. public class CarAutoConfiguration {  
3.     @Bean  
4.     @ConditionalOnBean  
5.     public CarService carService() { ... }  
6. }
```



D

```
1. @Configuration  
2. public class CarAutoConfiguration {  
3.     @Bean  
4.     @ConditionalOnMissingBean  
5.     public CarService carService() { ... }  
6. }
```

Question 7 of 11

Flag Jump to Qs 00:12:03 < PREV NEXT C RESET

Java: Basic String Manipulation

You are given a Java program to convert all the lowercase letters of a string to uppercase and vice versa. However, some parts of the code are either incomplete or contain errors. Your task is to fix the code and make it work as intended.

INSTRUCTIONS:

- Comments have been provided in the respective code areas to understand the problem better.

```

5: public class Main {
6:     public static void main(String[] args) throws IOException {
7:         BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
8:         BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getProperty("OUTPUT_FILE_PATH")));
9:         bufferedWriter.write("\n");
10:        bufferedWriter.close();
11:        bufferedWriter = new BufferedWriter(new FileWriter(System.getProperty("OUTPUT_FILE_PATH"), true));
12:        String str1 = bufferedReader.readLine();
13:
14:        StringBuffer newStr = new StringBuffer(str1);
15:
16:        for(int i = 0; i < str1.length(); i++) {
17:
18:            //Add a condition that checks if the character at "i" is lowercase
19:            if(Character.isLowerCase(str1.charAt(i))) {
20:                newStr.setCharAt(i, Character.toUpperCase(str1.charAt(i))) /*Call the function to convert the character*/
21:            }
22:
23:            //Add a condition that checks if the character at "i" is uppercase
24:            else if(Character.isUpperCase(str1.charAt(i))) {
25:                newStr.setCharAt(i, Character.toLowerCase(str1.charAt(i)))/*Call the function to convert the character*/
26:            }
27:        }
28:
29:        bufferedWriter.write(newStr.toString() + "\n");
30:        bufferedWriter.newLine();
31:
32:        bufferedReader.close();
33:        bufferedWriter.close();
34:    }
}

```

INPUT / OUTPUT Test with custom input RUN CODE RUN ALL TEST

9

Spring Boot provides logging functionality so that you can trace the application transactions back for security purposes. To achieve this, Spring Boot provides an API, which is known as Log4J. There are many levels of Log4J. Which are these levels?

(Select all that apply below)

A DEBUG

B Appear

C Correct

D WARN

E Excepnon

10

Topic: MongoDB

Complete the following code to find out how many views an author has for all the blogs.

(There are multiple drop-downs below. Please select a correct response for each drop-down)

```
db.runCommand( {
    mapReduce : "blogs",
    map : function() {
        emit( this.author_name, this.views );
    },
    reduce : function(name, counts) {
        var count = 0;
        for (var index  0; index < counts.length; ++index) {
            count += counts[index];
        }
        return count;
    },
    out: {
        inline: 1
    }
} )
```

11

(There are multiple drop-downs below. Select a correct response for each drop-down)

```
import org.apache.activemq.ActiveMQConnectionFactory;
import javax.jms.ConnectionFactory;
import javax.jms.Connection;
import javax.jms.Session;
import javax.jms.Destination;
import javax.jms.MessageProducer;
import javax.jms.TextMessage;

String destName = "some_dest";
ConnectionFactory connectionFactory =
     new ActiveMQConnectionFactory (
        ActiveMQConnection.DEFAULT_BROKER_URL);
Connection connection = connectionFactory.  createConnection();
connection.start();
Session session =
    connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
Destination destination = session.createQueue(destName);
MessageProducer messageProducer = session.createProducer(destination);
TextMessage message =  session  .createTextMessage("Hello!");
producer.send(message);
session.close();
connection.close();
}
```