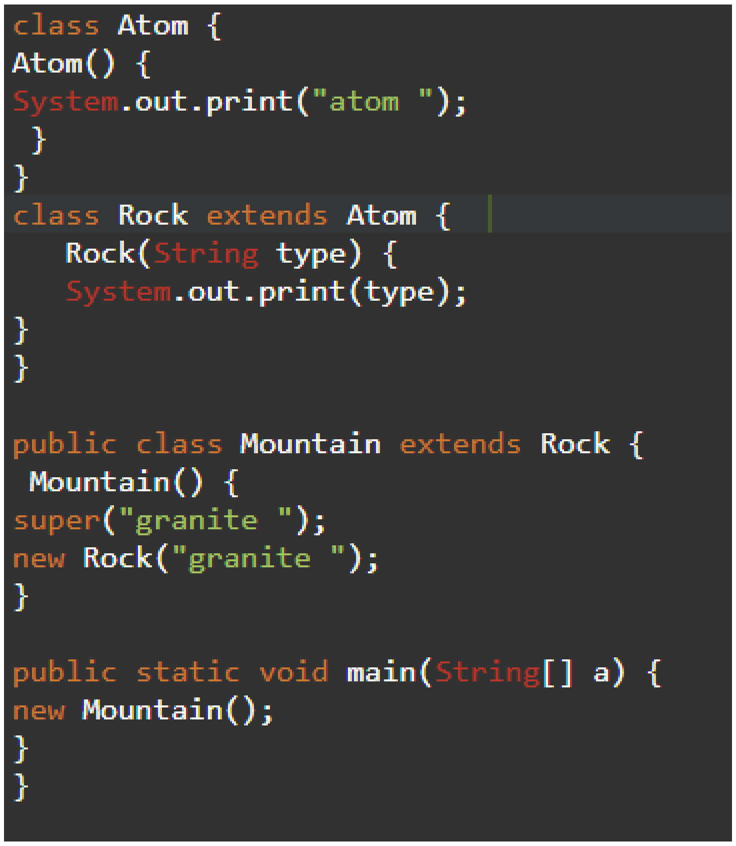
The Test is divided into three sections. The first section has MCQ's on Java, each question carries 2 marks, And the second section has Queries on MySQL, each question carries 3 marks. For the third section, three questions need to be completed, each question carries 25 marks.

Duration of the Test: 3 hrs.

Section-1: Core Java MCQ's

Q1)



```
class Atom {
Atom() {
System.out.print("atom ");
 }
}
class Rock extends Atom {
   Rock(String type) {
   System.out.print(type);
}
}

public class Mountain extends Rock {
 Mountain() {
super("granite ");
new Rock("granite ");
}

public static void main(String[] a) {
new Mountain();
}
}
```

What is the result?

A. Compilation fails.

B. atom granite

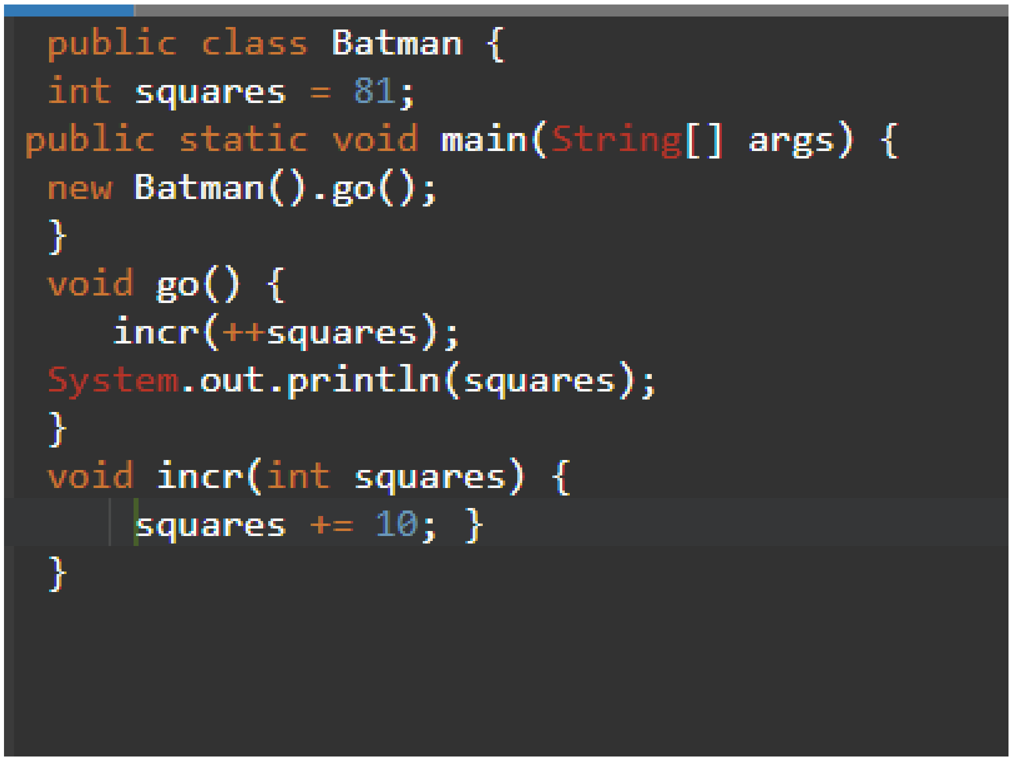
C. granite granite

D. atom granite granite

E. An exception is thrown at runtime.

F. atom granite atom granite

Q2)



```
public class Batman {
int squares = 81;
public static void main(String[] args) {
new Batman().go();
}
void go() {
    incr(++squares);
System.out.println(squares);
}
void incr(int squares) {
    squares += 10; }
}
```

What is the result?

A. 81

B. 82

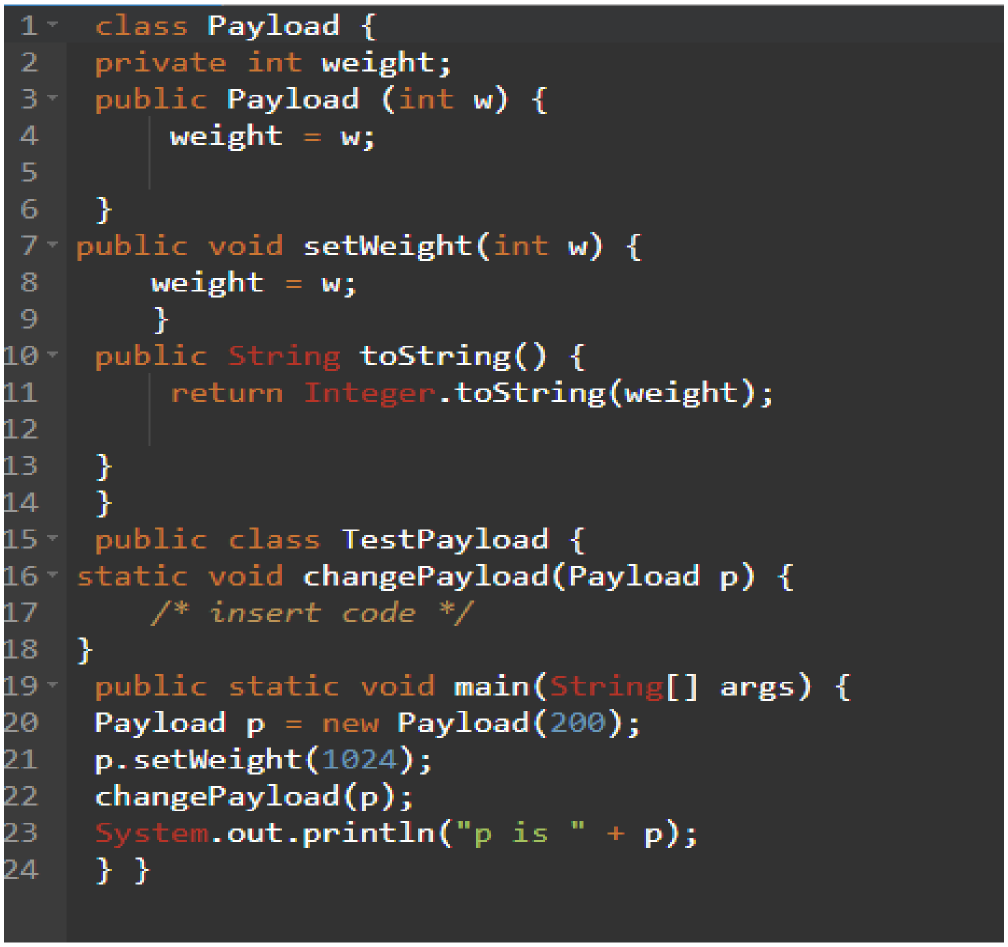
C. 91

D. 92

E. Compilation fails.

F. An exception is thrown at runtime.

Q3)



```
1   class Payload {
2   private int weight;
3   public Payload (int w) {
4       weight = w;
5
6   }
7  public void setWeight(int w) {
8      weight = w;
9      }
10  public String toString() {
11      return Integer.toString(weight);
12
13  }
14  }
15  public class TestPayload {
16 static void changePayload(Payload p) {
17     /* insert code */
18 }
19  public static void main(String[] args) {
20  Payload p = new Payload(200);
21  p.setWeight(1024);
22  changePayload(p);
23  System.out.println("p is " + p);
24  } }
```

Which code fragment, inserted in the line 17, produces the output p is 420?

A. p.setWeight(420);

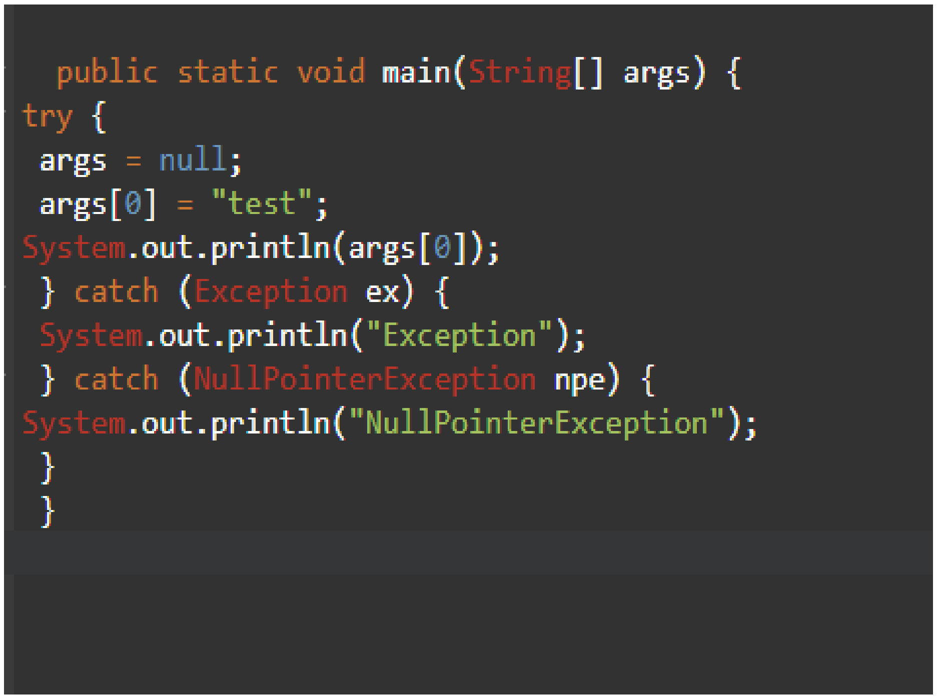
B. p.changePayload(420);

C. p = new Payload(420);

D. Payload.setWeight(420);

E. p = Payload.setWeight(420);

Q4)



```
  public static void main(String[] args) {
try {
 args = null;
 args[0] = "test";
System.out.println(args[0]);
 } catch (Exception ex) {
 System.out.println("Exception");
 } catch (NullPointerException npe) {
System.out.println("NullPointerException");
 }
 }
```

What is the result?

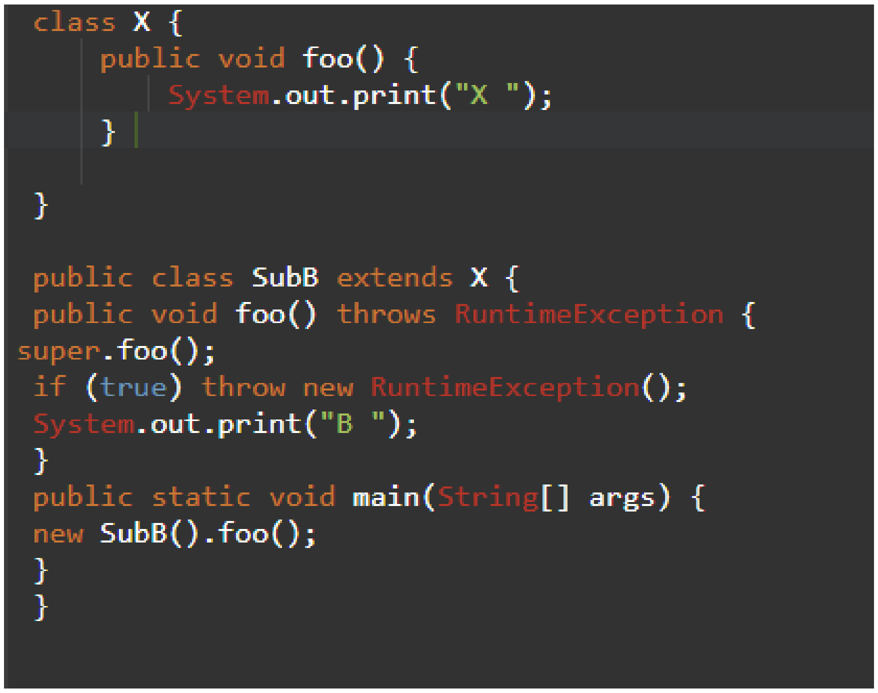
A. test

B. Exception

C. Compilation fails.

D. NullPointerException

Q5)



```
class X {
    public void foo() {
        System.out.print("X ");
    }

}

public class SubB extends X {
public void foo() throws RuntimeException {
super.foo();
if (true) throw new RuntimeException();
System.out.print("B ");
}
public static void main(String[] args) {
new SubB().foo();
}
}
```

What is the result?

A. X, followed by an Exception.

B. No output, and an Exception is thrown.

C. Compilation fails due to an error on line 14.

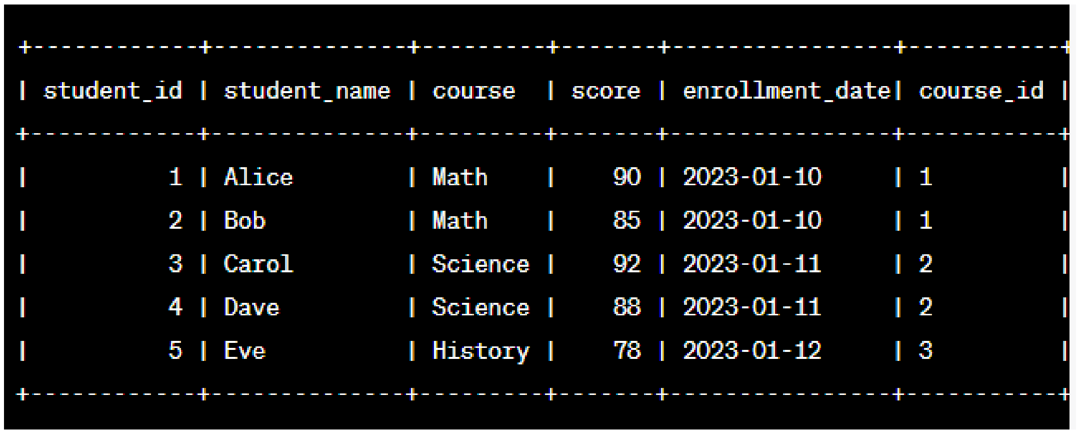
D. Compilation fails due to an error on line 16.

E. Compilation fails due to an error on line 17.

F. X, followed by an Exception, followed by B.

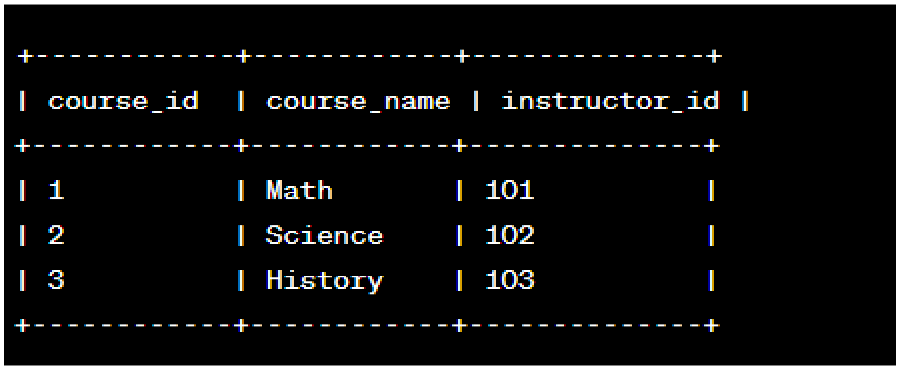
Section-2: MySQL Queries

Students Table:



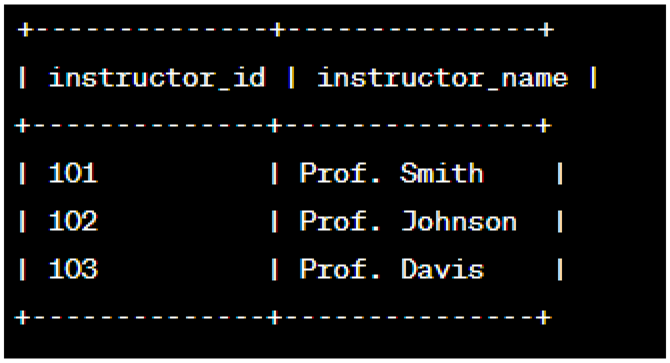
```
+------------+-------------+---------+-------+---------------+-----------+
| student_id | student_name | course  | score | enrollment_date| course_id |
+------------+-------------+---------+-------+---------------+-----------+
|          1 | Alice        | Math    |    90 | 2023-01-10     | 1         |
|          2 | Bob          | Math    |    85 | 2023-01-10     | 1         |
|          3 | Carol        | Science |    92 | 2023-01-11     | 2         |
|          4 | Dave         | Science |    88 | 2023-01-11     | 2         |
|          5 | Eve          | History |    78 | 2023-01-12     | 3         |
+------------+-------------+---------+-------+---------------+-----------+
```

Courses Table:



```
+------------+------------+--------------+
| course_id  | course_name | instructor_id |
+------------+------------+--------------+
| 1          | Math        | 101          |
| 2          | Science     | 102          |
| 3          | History     | 103          |
+------------+------------+--------------+
```

Instructors Table:


```
+--------------+---------------+
| instructor_id | instructor_name |
+--------------+---------------+
| 101          | Prof. Smith    |
| 102          | Prof. Johnson  |
| 103          | Prof. Davis    |
+--------------+---------------+
```

1.Find the course with the highest average score and the instructor who teaches it.

Ans:____

____

2.List students who have scores in a course that are higher than the average score of "Math" and "Science" courses.

Ans:____

____

3. List the courses that have the same instructor for two or more courses.

Ans:____

____

4.List the students who have not enrolled in any courses instructed by "Prof. Smith".

Ans:____

____

5. Find the course with the lowest average score among courses with at least two students.

Ans:____

____

Note: You can use the provided table data to solve the SQL queries for each of the given challenge.

Section-3: Coding challenges on core java concepts

Q1) You are working on a ticketing system for a popular concert. The system keeps track of available seats in different sections of the venue. A customer wants to purchase a ticket for the concert. The system needs to provide a list of available seats for the customer to choose from. You have two sorted arrays representing available seats in two different sections of the concert hall. You need to merge these arrays efficiently to provide a single sorted list of available seats for customers. To ensure a smooth booking experience, you must merge the two arrays efficiently.

Expected Input:

Array 1: [1, 3, 5]

Array 2: [2, 4, 6]

Expected Output:

Merged Sorted Array: [1, 2, 3, 4, 5, 6]

Q2) You have a list of numbers that represents data points from a sensor. You want to find the longest sequence of consecutive numbers in this list because it may indicate a significant pattern in the data.

You have a list of numbers:

- Data Points: [100, 4, 200, 1, 3, 2, 201, 202, 203, 204, 205, 5, 6,7]

Expected Input:

List of Numbers: [100, 4, 200, 1, 3, 2, 201, 202, 203, 204, 205, 5, 6,7]

Expected Output:

Longest Consecutive Sequence: [1, 2, 3, 4, 5, 6,7]

Expected Input:

List of Numbers: [22, 23, 24, 25, 26, 29, 30, 31, 32, 33, 35, 36, 38, 39, 40, 42]

Expected Output:

Longest Consecutive Sequence: [22, 23, 24, 25,26]

Expected Input:

List of Numbers:[50, 23, 89, 67, 1, 2, 67, 32, 3, 8, 4]
Expected Output:

Longest Consecutive Sequence:[1, 2, 3, 4]

Q3) Imagine you are building an autocompletion feature for a text editor or a search engine. Users start typing a word, and as they type, the system should suggest possible completions based on the common prefixes of words in the dictionary. Finding common prefixes can be useful in various applications, such as autocompletion or searching.Write a Java program to find and output the common prefix of these words.

Expected Input:

List of Strings: ["flower", "flow", "flight", "flock", "flat"]

Expected Output:

Common Prefix: "fl"

Q4) Imagine You are a language enthusiast who enjoys exploring the richness of various languages. You come across the concept of pangrams, which are sentences that contain every letter of the alphabet at least once. Intrigued by this concept, you decide to verify if a sentence you encountered in a book is indeed a pangram.

A pangram is a sentence that contains every letter of the alphabet at least once. write Java program is used to check whether a given sentence is a pangram, helping you appreciate the linguistic complexity of the sentence you encountered in your reading. It's an exploration of language and the concept of pangrams, which can be a fun linguistic exercise.

Expected Input:

Enter a sentence: "The quick brown fox jumps over the lazy dog."

Expected Output:

"The quick brown fox jumps over the lazy dog." is a pangram.

Expected Input:

Enter a sentence: "The lazy cat slept."

Expected Output:

"The lazy cat slept." is not a pangram.

Q5) You are part of a team developing a music streaming service that aims to provide an excellent music listening experience. This service illustrates how users of the music streaming service can create playlists, manage their music library, and enjoy a personalized music experience. The service allows users to create playlists, search for songs, and play music from their library. Write a Java program for a music streaming service. The program should allow users to create playlists, search for songs, and play music from their library. You need to design classes for songs, playlists, and implement the following functionalities:

Your program should provide the following functionality:

- Users can create playlists, add songs to their playlists, and view the contents of their playlists.

- Users can search for songs by title, artist, or genre and see search results.

- Users can play songs from their library.

Expected Interaction:

Welcome to the Music Streaming Service!

1. Create Playlist

2. Add Song to Playlist

3. Search for Songs

4. View My Playlists

5. Exit

Please select an option: 1

```
 Enter the name of your new playlist: My Favorites
Playlist "My Favorites" created.

Please select an option: 2
Enter the title of the song to add to your playlist: Imagine
Enter the artist: John Lennon
Enter the duration (in minutes): 3:00
Enter the genre: Rock
Song "Imagine" added to "My Favorites" playlist.

Please select an option: 2
Enter the title of the song to add to your playlist: Thriller
Enter the artist: Michael Jackson
Enter the duration (in minutes): 5:57
Enter the genre: Pop
Song "Thriller" added to "My Favorites" playlist.

Please select an option: 3
 Enter the search criteria (title, artist, or genre): artist
Enter the search term: John Lennon
Search Results:
1. Title: Imagine, Artist: John Lennon, Duration: 3:00, Genre: Rock


Please select an option: 4
 My Playlists:
1. My Favorites
 Please select a playlist to view: 1
 Playlist "My Favorites":
1. Title: Imagine, Artist: John Lennon, Duration: 3:00, Genre: Rock
2. Title: Thriller, Artist: Michael Jackson, Duration: 5:57, Genre: Pop

Please select an option: 5
Thank You(Exi
```