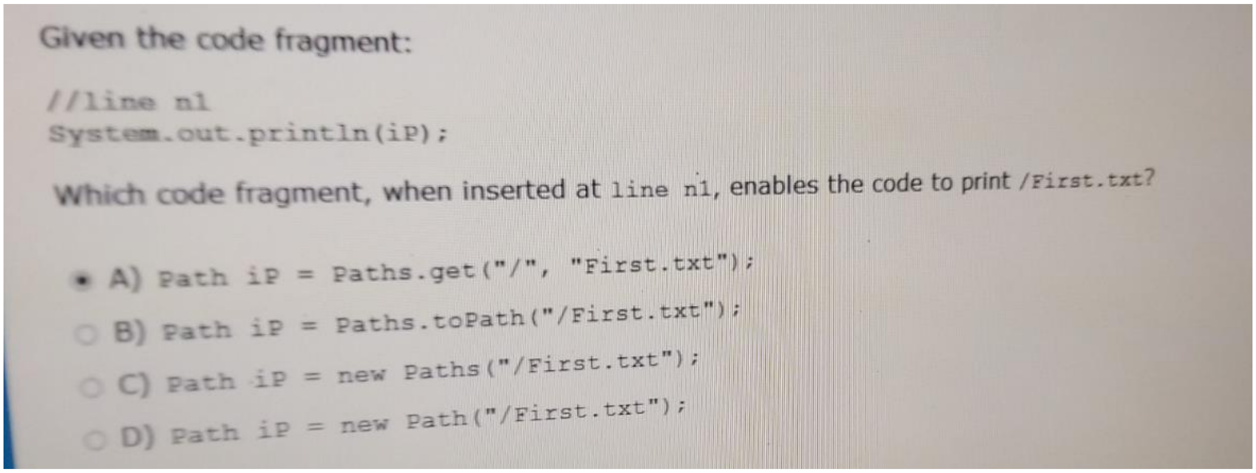
Problem 1:



Given the code fragment:

```
//line n1
System.out.println(iP);
```

Which code fragment, when inserted at `line n1`, enables the code to print `/First.txt`?

● A) `Path iP = Paths.get("/", "First.txt");`

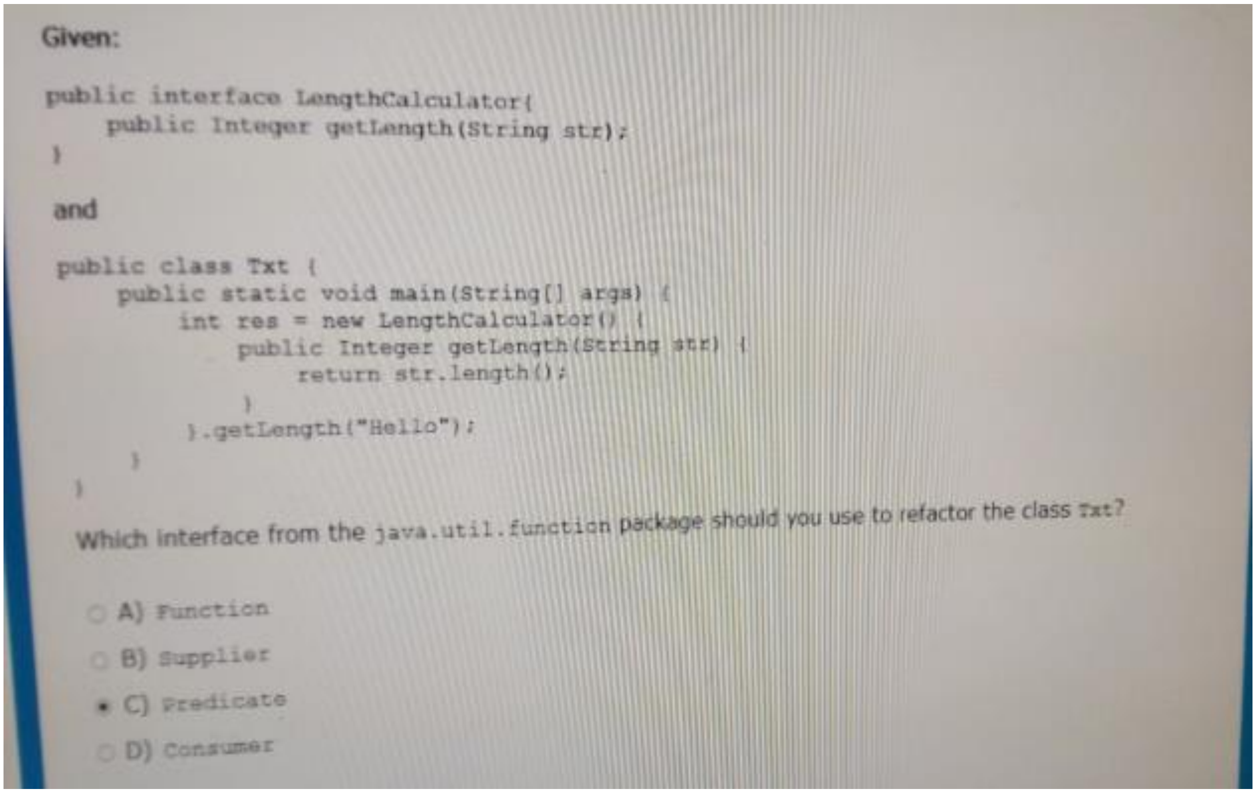
○ B) `Path iP = Paths.toPath("/First.txt");`

○ C) `Path iP = new Paths("/First.txt");`

○ D) `Path iP = new Path("/First.txt");`

Correct Answer: A (Note it is wrong question, but closest Answer is A. it will work when we use something like Path iP = Paths.get("//ok/", "First.txt");

Problem 2:



Given:

```
public interface LengthCalculator{
    public Integer getLength(String str);
}
```

and

```
public class Txt {
    public static void main(String[] args) {
        int res = new LengthCalculator() {
            public Integer getLength(String str) {
                return str.length();
            }
        }.getLength("Hello");
    }
}
```

Which interface from the `java.util.function` package should you use to refactor the class `Txt`?

○ A) `Function`

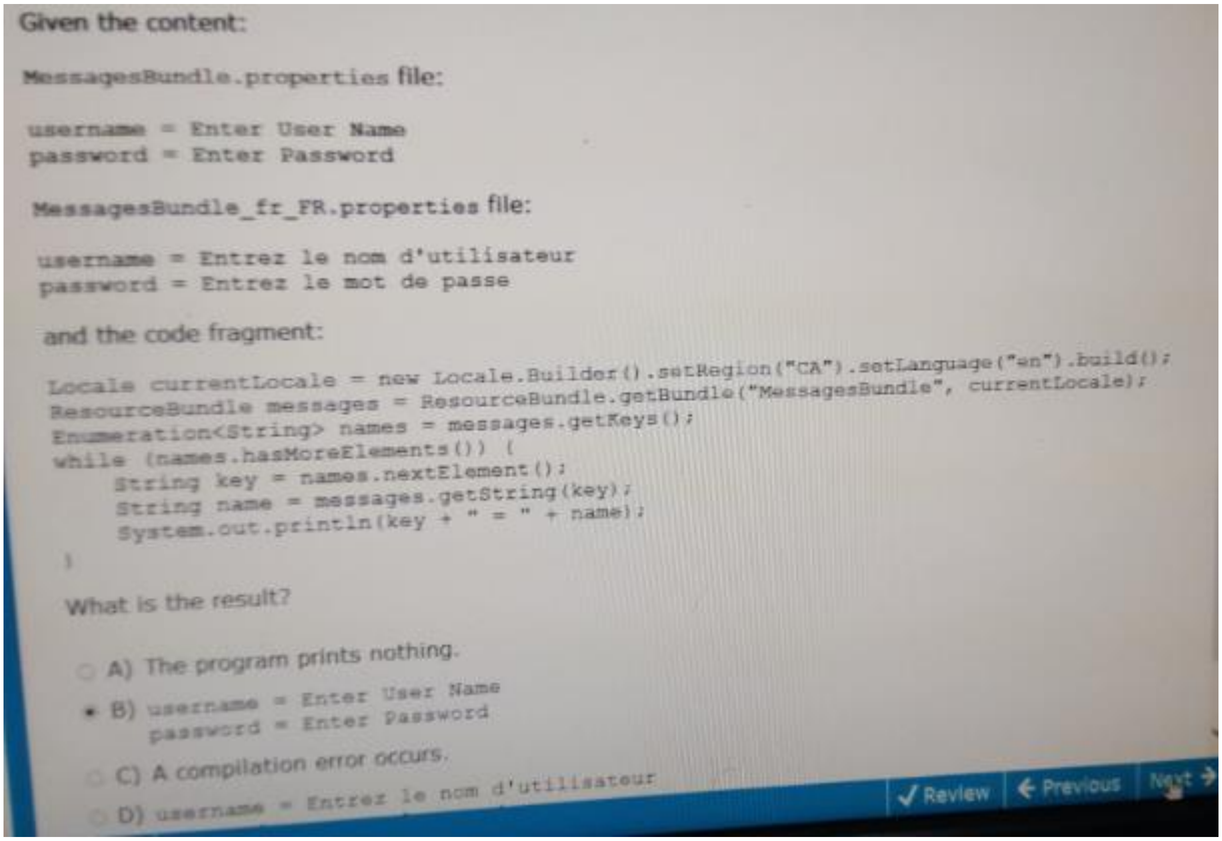
○ B) `Supplier`

● C) `Predicate`

○ D) `Consumer`

Correct Answer: C

Problem 3:



Given the content:

`MessagesBundle.properties` file:

```
username = Enter User Name
password = Enter Password
```

`MessagesBundle_fr_FR.properties` file:

```
username = Entrez le nom d'utilisateur
password = Entrez le mot de passe
```

and the code fragment:

```
Locale currentLocale = new Locale.Builder().setRegion("CA").setLanguage("en").build();
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
Enumeration<String> names = messages.getKeys();
while (names.hasMoreElements()) {
    String key = names.nextElement();
    String name = messages.getString(key);
    System.out.println(key + " = " + name);
}
```

What is the result?

○ A) The program prints nothing.

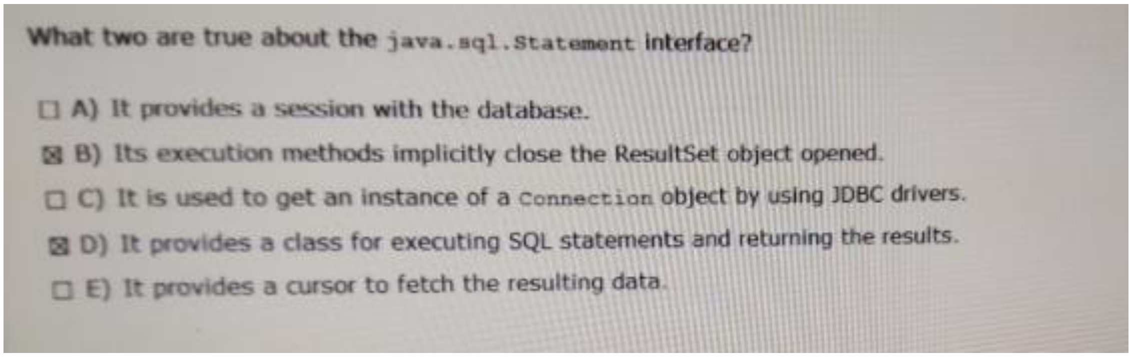
● B) `username = Enter User Name`
`password = Enter Password`

○ C) A compilation error occurs.

○ D) `username = Entrez le nom d'utilisateur`

Correct Answer:

Problem 4:



What two are true about the `java.sql.Statement` interface?

☐ A) It provides a session with the database.

☒ B) Its execution methods implicitly close the ResultSet object opened.

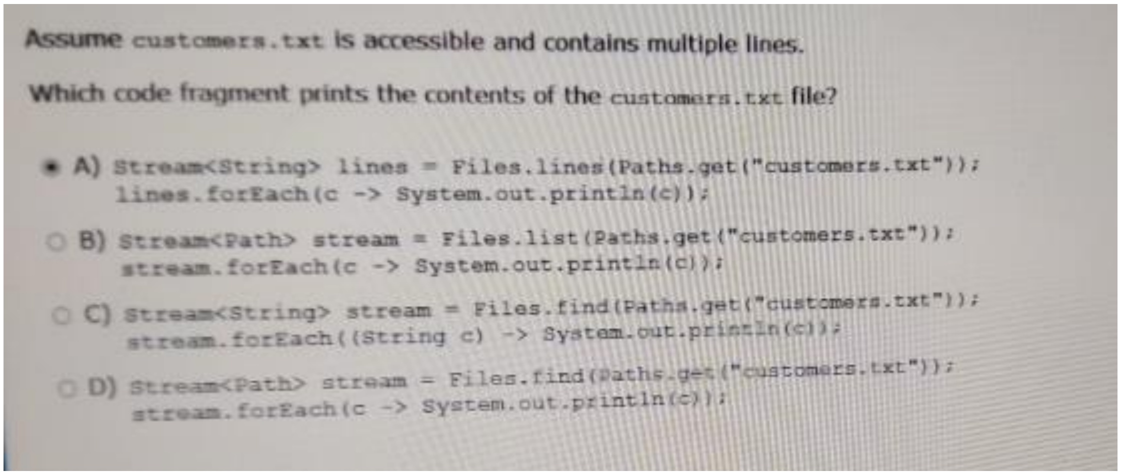
☐ C) It is used to get an instance of a `Connection` object by using JDBC drivers.

☒ D) It provides a class for executing SQL statements and returning the results.

☐ E) It provides a cursor to fetch the resulting data.

Correct Answer: D

Problem 5:



Assume `customers.txt` is accessible and contains multiple lines.

Which code fragment prints the contents of the `customers.txt` file?

● A)
```
Stream<String> lines = Files.lines(Paths.get("customers.txt"));
lines.forEach(c -> System.out.println(c));
```

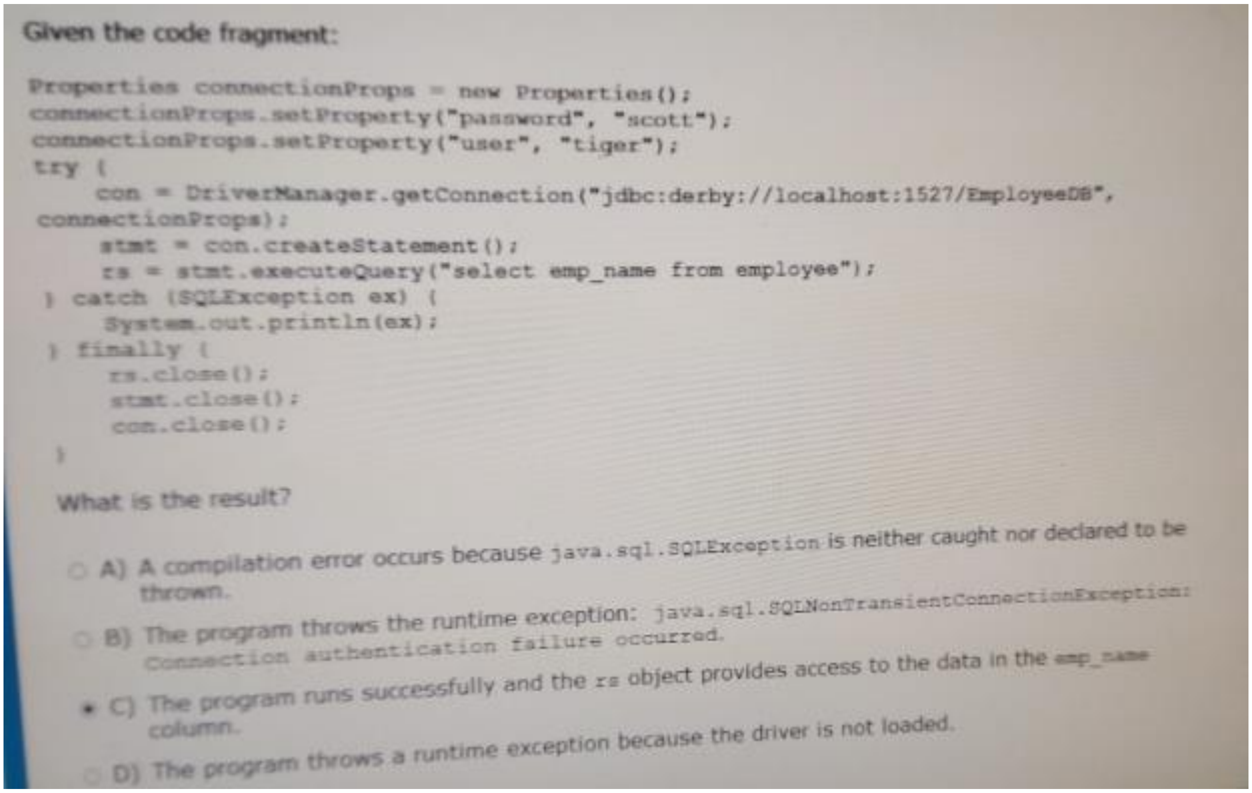
○ B)
```
Stream<Path> stream = Files.list(Paths.get("customers.txt"));
stream.forEach(c -> System.out.println(c));
```

○ C)
```
Stream<String> stream = Files.find(Paths.get("customers.txt"));
stream.forEach((String c) -> System.out.println(c));
```

○ D)
```
Stream<Path> stream = Files.find(Paths.get("customers.txt"));
stream.forEach(c -> System.out.println(c));
```

Correct Answer: A

Problem 6:



Given the code fragment:

```
Properties connectionProps = new Properties();
connectionProps.setProperty("password", "scott");
connectionProps.setProperty("user", "tiger");
try {
    con = DriverManager.getConnection("jdbc:derby://localhost:1527/EmployeeDB",
connectionProps);
    stmt = con.createStatement();
    rs = stmt.executeQuery("select emp_name from employee");
} catch (SQLException ex) {
    System.out.println(ex);
} finally {
    rs.close();
    stmt.close();
    con.close();
}
```

What is the result?

○ A) A compilation error occurs because `java.sql.SQLException` is neither caught nor declared to be thrown.

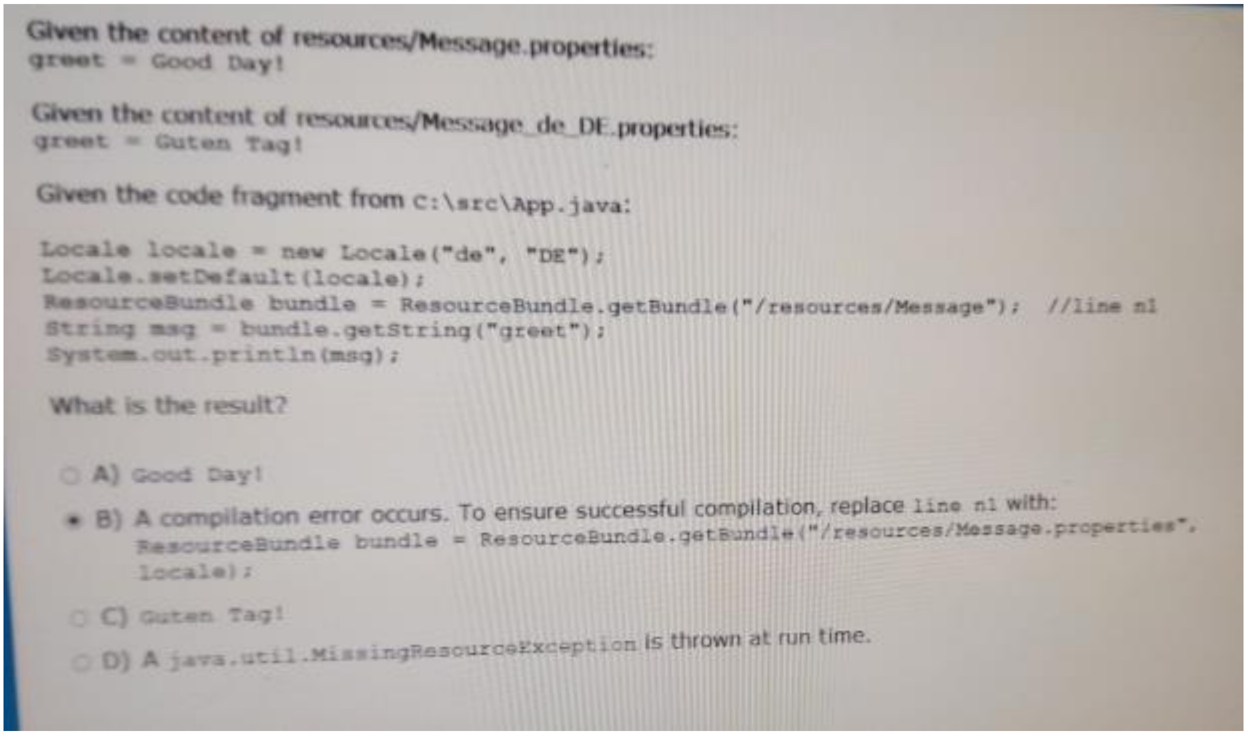
○ B) The program throws the runtime exception: `java.sql.SQLNonTransientConnectionException: Connection authentication failure occurred.`

● C) The program runs successfully and the `rs` object provides access to the data in the `emp_name` column.

○ D) The program throws a runtime exception because the driver is not loaded.

Correct Answer:

Problem 7:



Given the content of resources/Message.properties:

```
greet = Good Day!
```

Given the content of resources/Message_de_DE.properties:

```
greet = Guten Tag!
```

Given the code fragment from `C:\src\App.java`:

```
Locale locale = new Locale("de", "DE");
Locale.setDefault(locale);
ResourceBundle bundle = ResourceBundle.getBundle("/resources/Message");  //line n1
String msg = bundle.getString("greet");
System.out.println(msg);
```

What is the result?

○ A) `Good Day!`

● B) A compilation error occurs. To ensure successful compilation, replace `line n1` with:

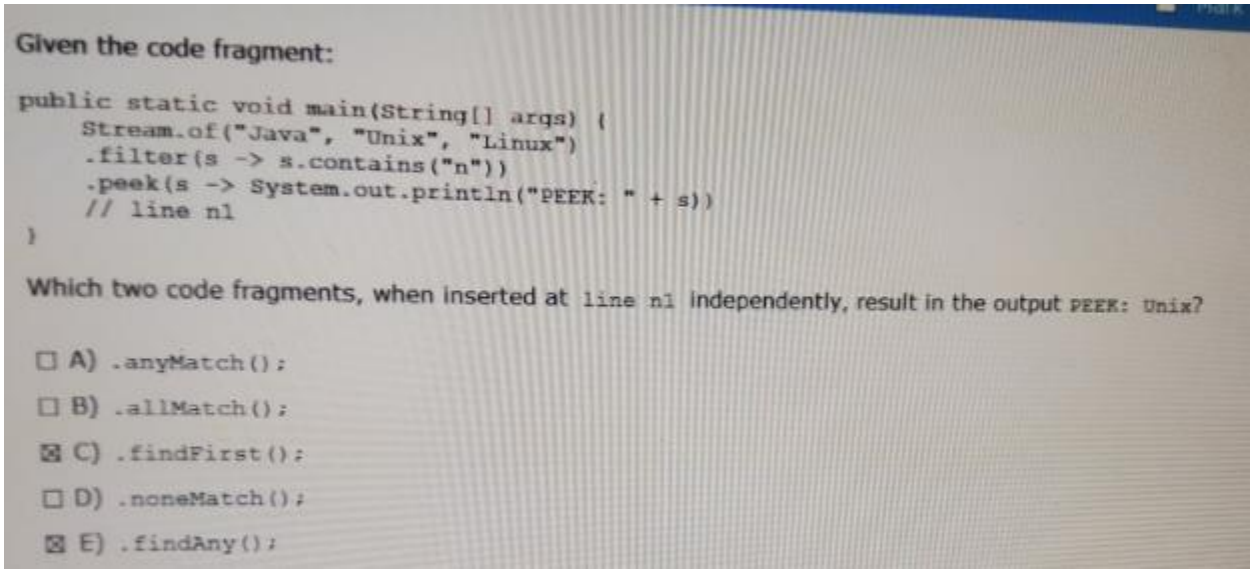
```
ResourceBundle bundle = ResourceBundle.getBundle("/resources/Message.properties",
locale);
```

○ C) `Guten Tag!`

○ D) A `java.util.MissingResourceException` is thrown at run time.

Correct Answer:

Problem 8:



Given the code fragment:

```
public static void main(String[] args) {
     Stream.of("Java", "Unix", "Linux")
     .filter(s -> s.contains("n"))
     .peek(s -> System.out.println("PEEK: " + s))
     // line n1
}
```

Which two code fragments, when inserted at `line n1` independently, result in the output `PEEK: Unix`?

☐ A) `.anyMatch();`

☐ B) `.allMatch();`

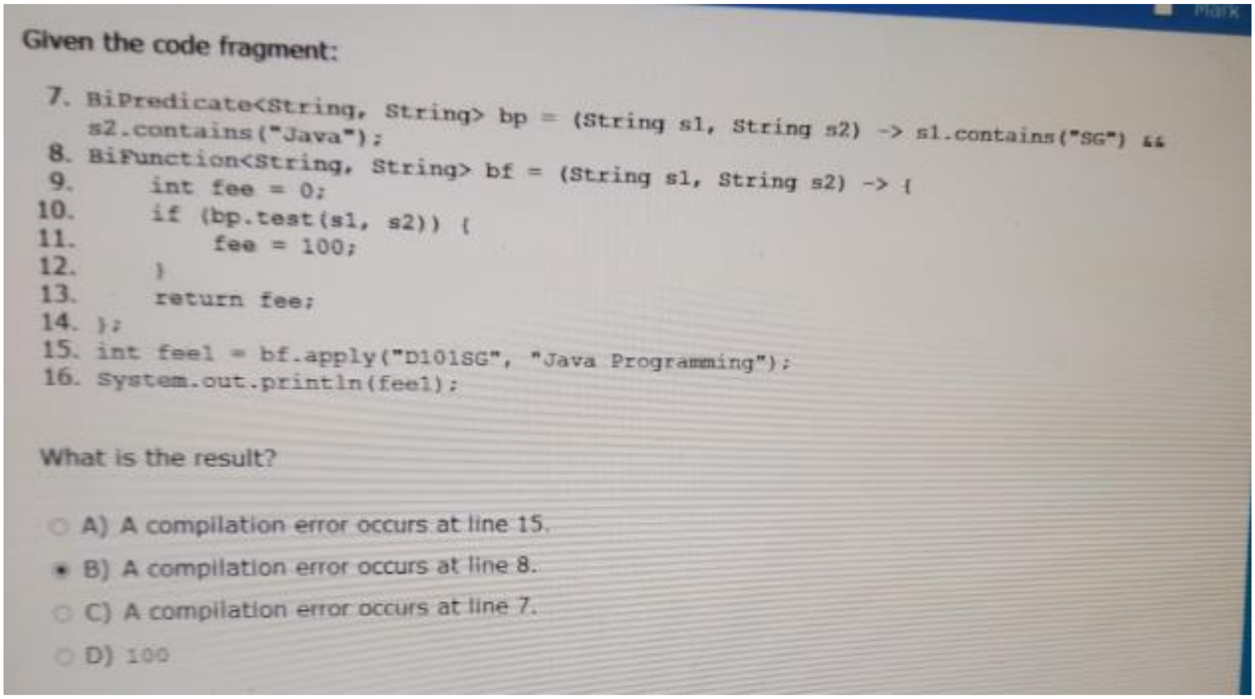
☒ C) `.findFirst();`

☐ D) `.noneMatch();`

☒ E) `.findAny();`

Correct Answer: C, E

Problem 9:



Given the code fragment:

```
7. BiPredicate<String, String> bp = (String s1, String s2) -> s1.contains("SG") &&
   s2.contains("Java");
8. BiFunction<String, String> bf = (String s1, String s2) -> {
9.       int fee = 0;
10.      if (bp.test(s1, s2)) {
11.          fee = 100;
12.      }
13.      return fee;
14. };
15. int fee1 = bf.apply("D101SG", "Java Programming");
16. System.out.println(fee1);
```

What is the result?

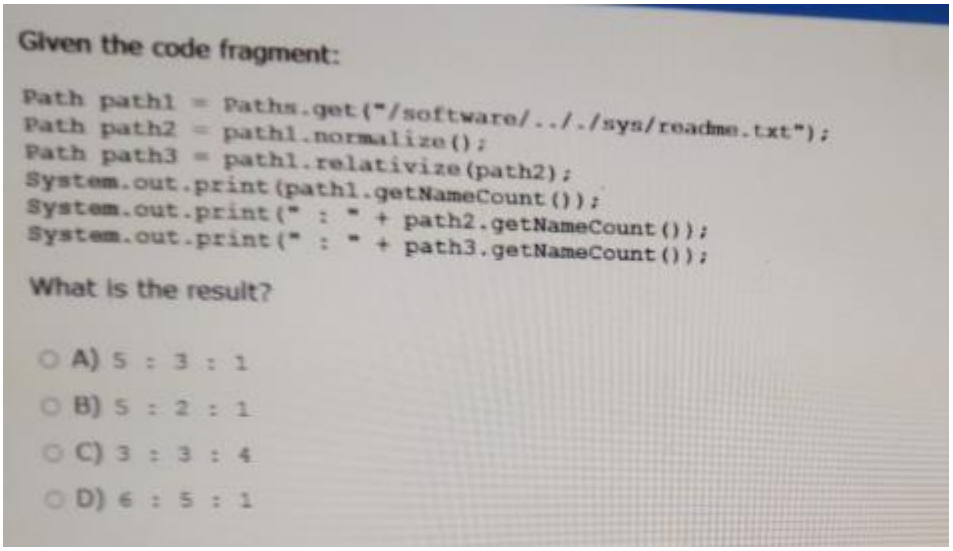
○ A) A compilation error occurs at line 15.

● B) A compilation error occurs at line 8.

○ C) A compilation error occurs at line 7.

○ D) 100

Problem 10:



Given the code fragment:

```
Path path1 = Paths.get("/software/.././sys/readme.txt");
Path path2 = path1.normalize();
Path path3 = path1.relativize(path2);
System.out.print(path1.getNameCount());
System.out.print(" : " + path2.getNameCount());
System.out.print(" : " + path3.getNameCount());
```

What is the result?

○ A) 5 : 3 : 1

○ B) 5 : 2 : 1

○ C) 3 : 3 : 4

○ D) 6 : 5 : 1

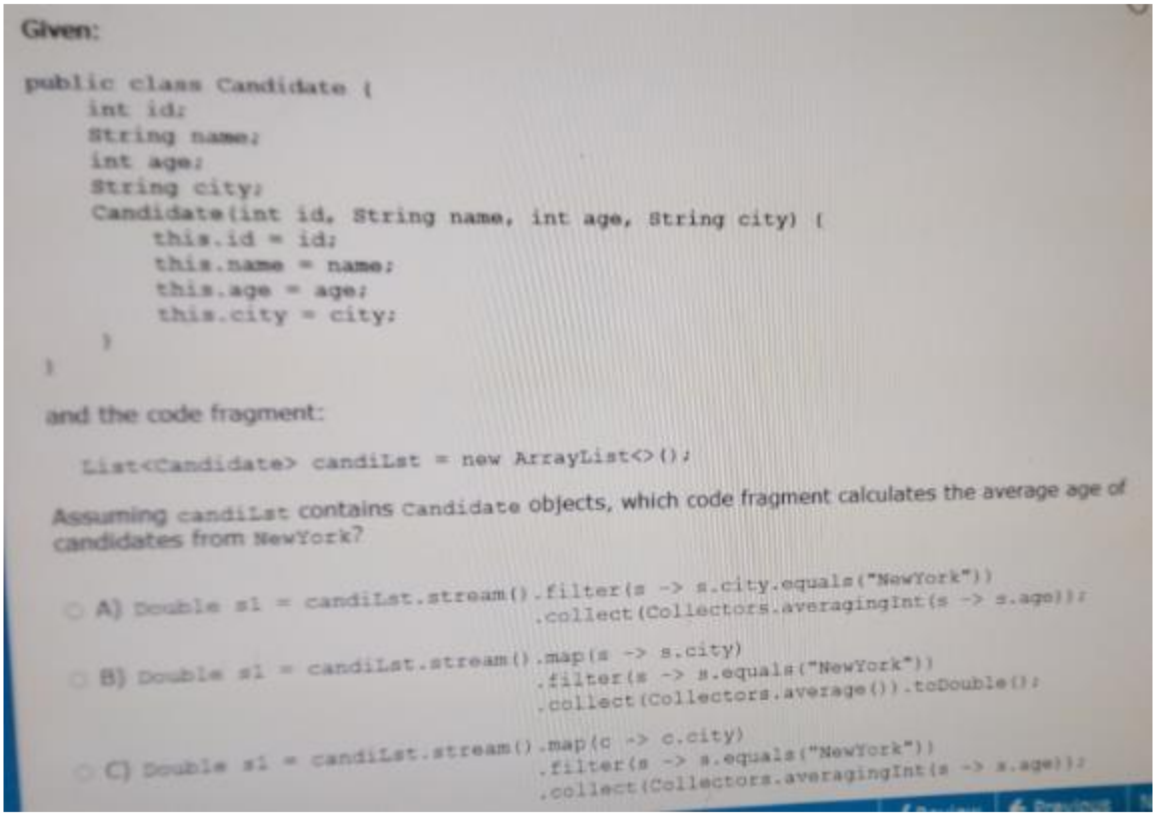
package q10;

import java.nio.file.Path;

```
import java.nio.file.Paths;


/**
 *
 * @author sami
 */
public class NewClass {


    public static void main(String[] args) {
        Path path1 = (Path) Paths.get("/software/../././sys/reademe.txt");
        Path path2 = path1.normalize();
        Path path3 = path1.relativize(path2);
        System.out.print(path1.getNameCount());
        System.out.print(" : " + path2.getNameCount());
        System.out.print(" : " + path3.getNameCount());


    }


}
```

Correct Answer is 5 : 2: 7, But Here nearest answer is B

Problem 11:



Given:

```
public class Candidate {
    int id;
    String name;
    int age;
    String city;
    Candidate(int id, String name, int age, String city) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.city = city;
    }
}
```

and the code fragment:

```
List<Candidate> candiLst = new ArrayList<>();
```

Assuming candiLst contains Candidate objects, which code fragment calculates the average age of candidates from NewYork?

○ A)
```
Double s1 = candiLst.stream().filter(s -> s.city.equals("NewYork"))
                             .collect(Collectors.averagingInt(s -> s.age));
```

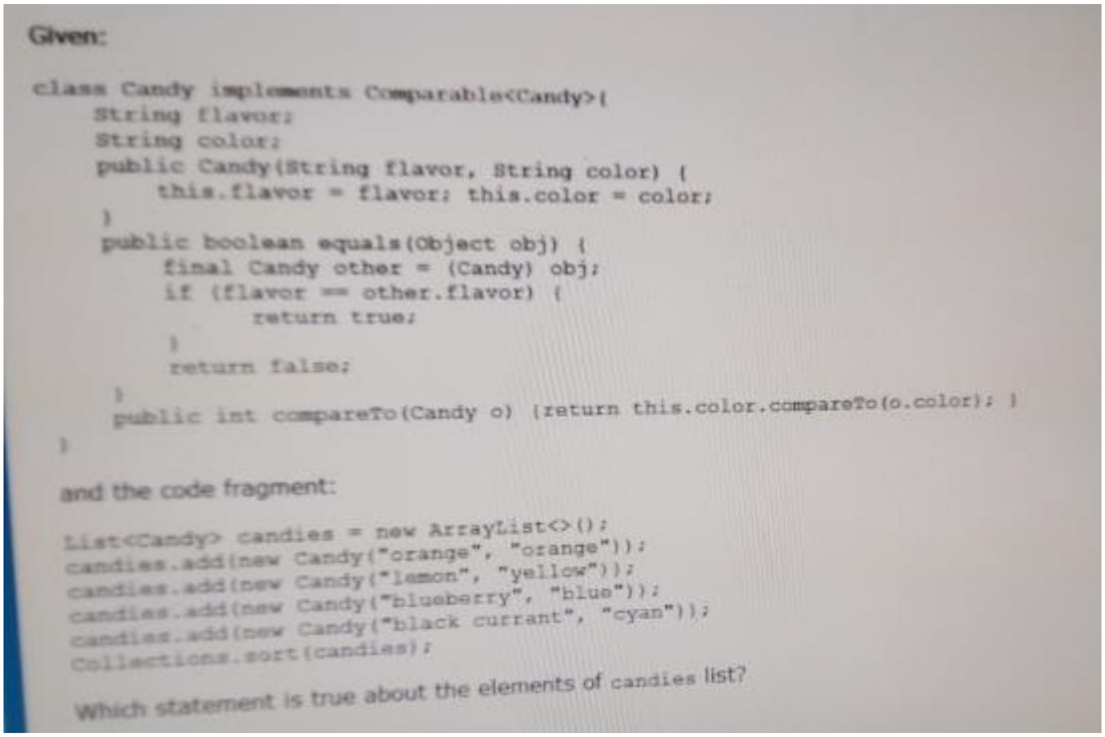
○ B)
```
Double s1 = candiLst.stream().map(s -> s.city)
                             .filter(s -> s.equals("NewYork"))
                             .collect(Collectors.average()).toDouble();
```

○ C)
```
Double s1 = candiLst.stream().map(c -> c.city)
                             .filter(s -> s.equals("NewYork"))
                             .collect(Collectors.averagingInt(s -> s.age));
```

Correct Answer: A

```
package com.coderbd.Q11;


import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;


//Ans: A
public class Candidate {
   int id;
   String name;
   int age;
   String city;


   public Candidate(int id, String name, int age, String city) {
    this.id = id;

    this.name = name;

    this.age = age;

    this.city = city;

  }

  public static void main(String[] args) {

    List<Candidate> candiList=new ArrayList<>();

   //A/// Double s1=candiList.stream().filter(s ->
s.city.equals("NewYork")).collect(Collectors.averagingInt(s -> s.age));

//       System.out.println(s1);

 //B//      Double s1=candiList.stream().map(mapper s -> s.city).filter(s -> s.equals("Newyork"));

//   C//     Double s1=candiList.stream().map( s ->s.city).filter(s-
>s.equals("Newyork")).collect(Collectors.average()).toDouble();

//D/ Double s1=candiList.stream().filter(s->s.equals("NewYoork")).map(c->c.age).average();

  }

}
```

Problem 12:



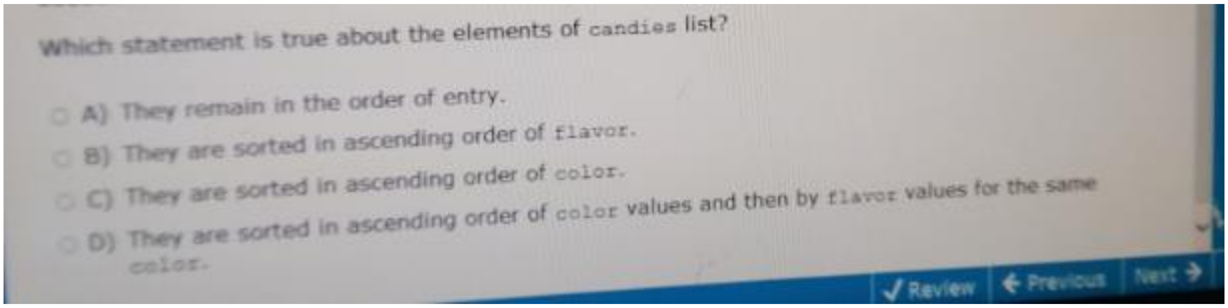
Given:

```
class Candy implements Comparable<Candy>{
    String flavor;
    String color;
    public Candy(String flavor, String color) {
        this.flavor = flavor; this.color = color;
    }
    public boolean equals(Object obj) {
        final Candy other = (Candy) obj;
        if (flavor == other.flavor) {
            return true;
        }
        return false;
    }
    public int compareTo(Candy o) {return this.color.compareTo(o.color); }
}
```

and the code fragment:

```
List<Candy> candies = new ArrayList<>();
candies.add(new Candy("orange", "orange"));
candies.add(new Candy("lemon", "yellow"));
candies.add(new Candy("blueberry", "blue"));
candies.add(new Candy("black currant", "cyan"));
Collections.sort(candies);
```

Which statement is true about the elements of candies list?



Which statement is true about the elements of candies list?

○ A) They remain in the order of entry.

○ B) They are sorted in ascending order of flavor.

○ C) They are sorted in ascending order of color.

○ D) They are sorted in ascending order of color values and then by flavor values for the same color.

Correct Answer: C

Look at Result: [Candy{falovour=blueberry, color=blue}, Candy{falovour=black corrent, color=cyan}, Candy{falovour=orange, color=orange}, Candy{falovour=lemon, color=yellow}]

package com.coderbd.Q12;

import java.util.ArrayList;

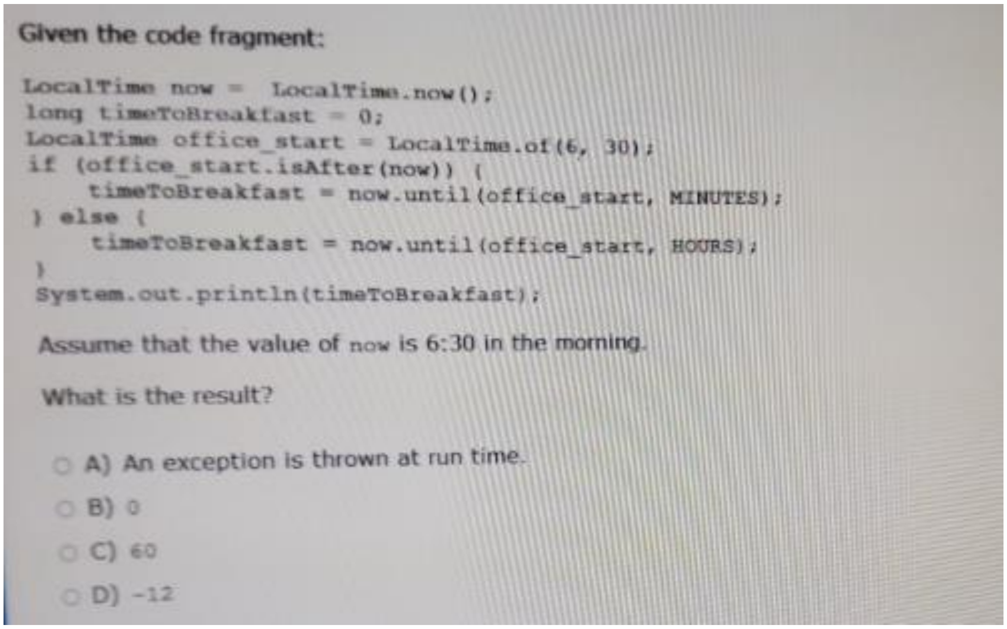
import java.util.Collections;

import java.util.List;

/**

```
 *
 * @author Touhid
 */
public class Candy implements Comparable<Candy> {

    String falovour;
    String color;

     public Candy(String falovour, String color) {
         this.falovour = falovour;
         this.color = color;
     }
    @Override
    public boolean equals (Object obj){
    final Candy other=(Candy)(Candy) obj;
        String flavour = null;
        //boolean flavour = false;
    if (flavour==other.falovour){
     return true;
}
    return false;
    }
    @Override
    public int compareTo(Candy o){
        return this.color.compareTo(o.color);
    }
     public static void main(String[] args) {
         List<Candy> candies= new ArrayList<>();
         candies.add(new Candy("orange","orange"));
         candies.add(new Candy("lemon","yellow"));

         candies.add(new Candy("blueberry","blue"));

         candies.add(new Candy("black corrent","cyan"));

         Collections.sort(candies);

         System.out.println(candies);


     }

     @Override

     public String toString() {

         return "Candy{" + "falovour=" + falovour + ", color=" + color + '}';

     }

}
```

Problem 13:



Given the code fragment:

```
LocalTime now =  LocalTime.now();
long timeToBreakfast = 0;
LocalTime office_start = LocalTime.of(6, 30);
if (office_start.isAfter(now)) {
     timeToBreakfast = now.until(office_start, MINUTES);
} else {
     timeToBreakfast = now.until(office_start, HOURS);
}
System.out.println(timeToBreakfast);
```

Assume that the value of `now` is 6:30 in the morning.

What is the result?

○ A) An exception is thrown at run time.

○ B) 0

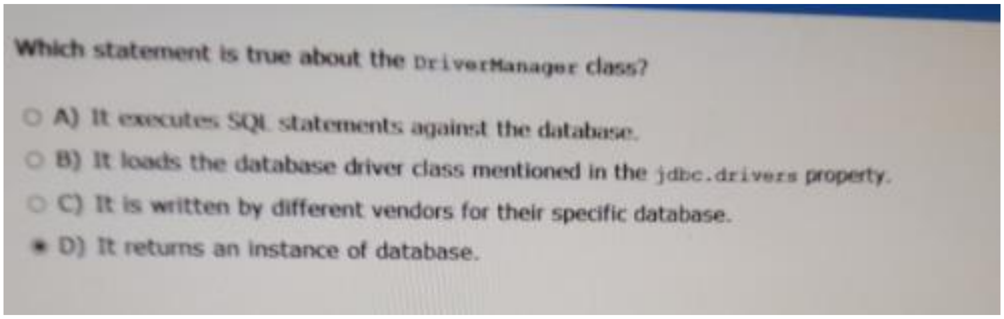
○ C) 60

○ D) -12

Correct Ans. B

```
package com.coderbd.Q13;


import java.time.LocalTime;

import static java.time.temporal.ChronoUnit.HOURS;

import static java.time.temporal.ChronoUnit.MINUTES;

import java.util.concurrent.TimeUnit;


public class Test {


    public static void main(String[] args) {

        LocalTime now = LocalTime.of(5, 50);;// assume now is 6 : 30

        long timeToBreakfast = 0;

        LocalTime office_start = LocalTime.of(6, 30);

        if (office_start.isAfter(now)) {

            timeToBreakfast = now.until(office_start, MINUTES);

        } else {
            timeToBreakfast = now.until(office_start, HOURS);
        }
        System.out.println(timeToBreakfast);

    }
    public static TimeUnit MINUTES1;

}
```

Problem 14:

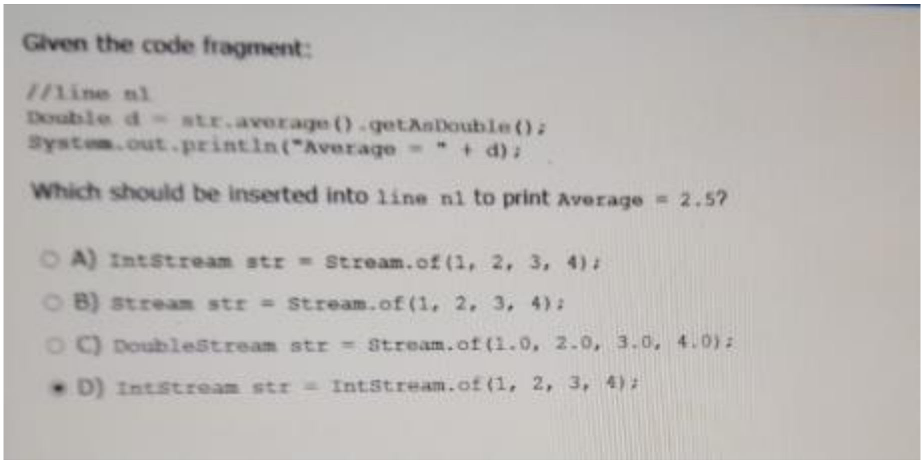


Which statement is true about the `DriverManager` class?

○ A) It executes SQL statements against the database.
○ B) It loads the database driver class mentioned in the `jdbc.drivers` property.
○ C) It is written by different vendors for their specific database.
● D) It returns an instance of database.

Correct Answer: D, Actually It returns an instance of Connection.

Problem 15:



Given the code fragment:

```
//line n1
Double d = str.average().getAsDouble();
System.out.println("Average = " + d);
```

Which should be inserted into `line n1` to print `Average = 2.5`?

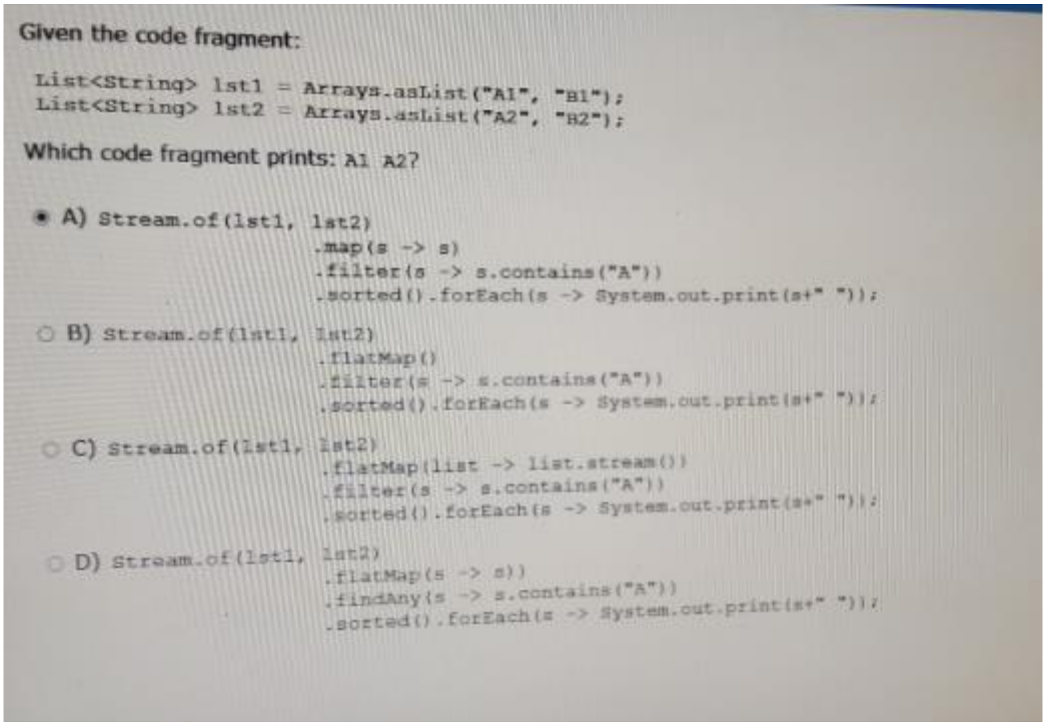
○ A) `IntStream str = Stream.of(1, 2, 3, 4);`
○ B) `Stream str = Stream.of(1, 2, 3, 4);`
○ C) `DoubleStream str = Stream.of(1.0, 2.0, 3.0, 4.0);`
● D) `IntStream str = IntStream.of(1, 2, 3, 4);`

Correct Answer: D

package com.coderbd.Q15;

```
import java.util.stream.IntStream;
import java.util.stream.Stream;

/**
 *
 * @author Touhid
 */
//Ans:D
public class Test {
    public static void main(String[] args) {
        //line n1
//        IntStream str=Stream.of(1,2,3,4);
//        Stream str=Stream.of(1,2,3,4);
//        DoubleStream str=Stream.of(1,2,3,4);
//        DoubleStream str=Stream.of(1,2,3,4);

IntStream str=IntStream.of(1,2,3,4);

        Double d=str.average().getAsDouble();
        System.out.println("Average"+d);

    }

}
```

Problem 16:



Given the code fragment:

```
List<String> lst1 = Arrays.asList("A1", "B1");
List<String> lst2 = Arrays.asList("A2", "B2");
```

Which code fragment prints: A1 A2?

● A)
```
Stream.of(lst1, lst2)
                .map(s -> s)
                .filter(s -> s.contains("A"))
                .sorted().forEach(s -> System.out.print(s+" "));
```

○ B)
```
Stream.of(lst1, lst2)
                .flatMap()
                .filter(s -> s.contains("A"))
                .sorted().forEach(s -> System.out.print(s+" "));
```

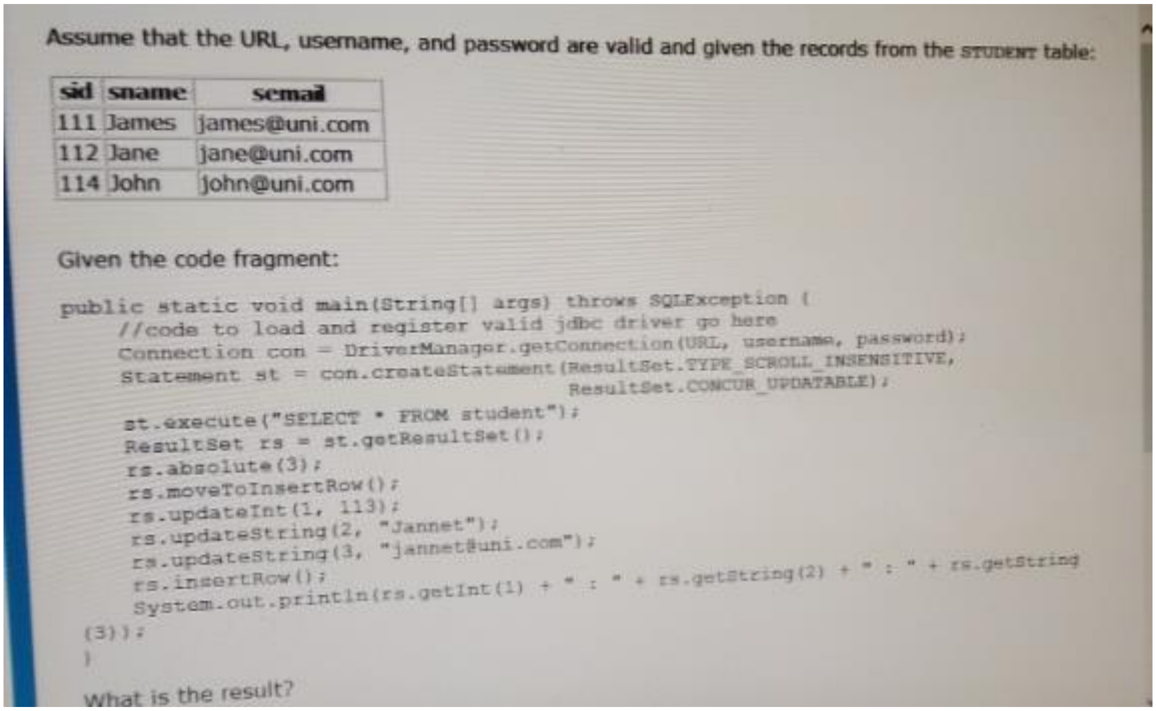
○ C)
```
Stream.of(lst1, lst2)
                .flatMap(list -> list.stream())
                .filter(s -> s.contains("A"))
                .sorted().forEach(s -> System.out.print(s+" "));
```

○ D)
```
Stream.of(lst1, lst2)
                .flatMap(s -> s))
                .findAny(s -> s.contains("A"))
                .sorted().forEach(s -> System.out.print(s+" "));
```

Correct Answer: C

```
package com.coderbd.Q16;


import java.util.Arrays;

import java.util.List;

import java.util.stream.Stream;


/**
 *
 * @author Touhid
 */

public class Test {

    public static void main(String[] args) {

        List<String> lst1=Arrays.asList("A1","B1");

        List<String> lst2=Arrays.asList("A2","B1");


// A       Stream.of(lst1,lst2)

//              .map(s->s)

//              .filter(s->s.contains("A"))

//              .sorted().forEach(s-> System.out.print(s+" "));

//   B     Stream.of(lst1,lst2)

//              .flatMap()

//              .filter(s->s.contains("A"))

//              .sorted().forEach(s-> System.out.print(s+" "));

                Stream.of(lst1,lst2)

                .flatMap(list->list.stream())

                .filter(s->s.contains("A"))

                .sorted().forEach(s-> System.out.print(s+" "));


    }


}
```

Problem 17:



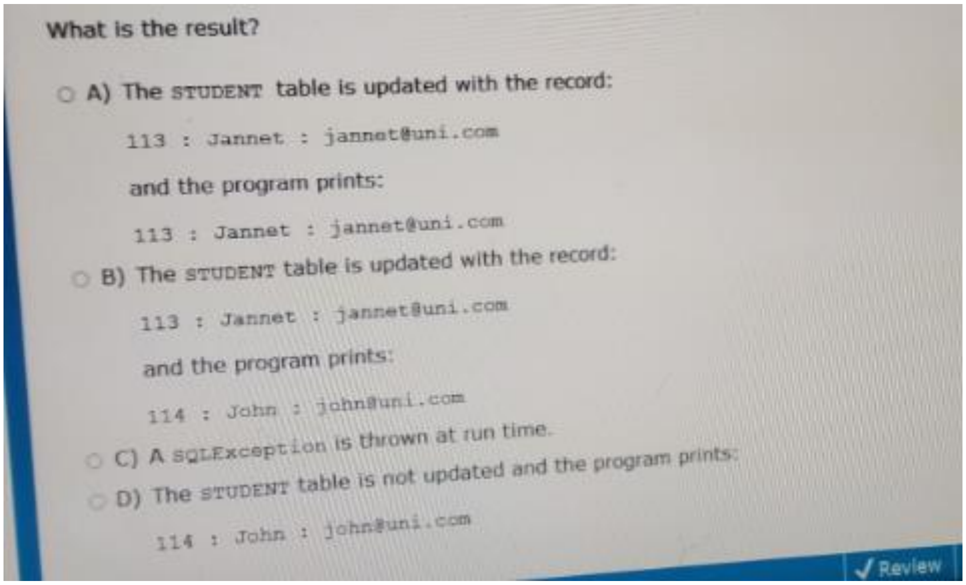
Assume that the URL, username, and password are valid and given the records from the STUDENT table:

sid	sname	semail
111	James	james@uni.com
112	Jane	jane@uni.com
114	John	john@uni.com

Given the code fragment:

```
public static void main(String[] args) throws SQLException {
    //code to load and register valid jdbc driver go here
    Connection con = DriverManager.getConnection(URL, username, password);
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                       ResultSet.CONCUR_UPDATABLE);

    st.execute("SELECT * FROM student");
    ResultSet rs = st.getResultSet();
    rs.absolute(3);
    rs.moveToInsertRow();
    rs.updateInt(1, 113);
    rs.updateString(2, "Jannet");
    rs.updateString(3, "jannet@uni.com");
    rs.insertRow();
    System.out.println(rs.getInt(1) + " : " + rs.getString(2) + " : " + rs.getString
(3));
}
```

What is the result?



What is the result?

○ A) The STUDENT table is updated with the record:

```
113 : Jannet : jannet@uni.com
```

and the program prints:

```
113 : Jannet : jannet@uni.com
```

○ B) The STUDENT table is updated with the record:

```
113 : Jannet : jannet@uni.com
```

and the program prints:

```
114 : John : john@uni.com
```

○ C) A SQLException is thrown at run time.

○ D) The STUDENT table is not updated and the program prints:

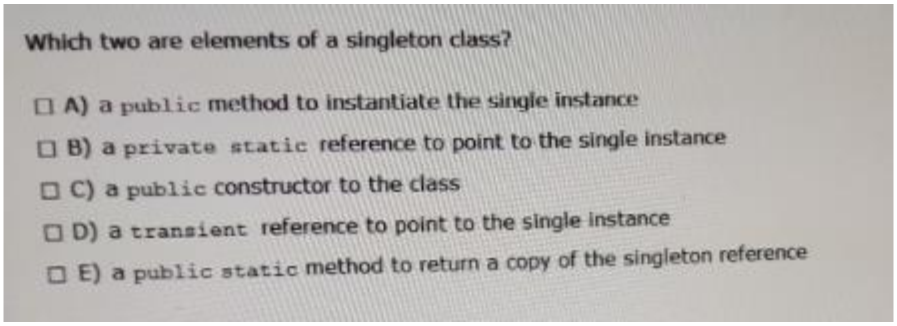
```
114 : John : john@uni.com
```

Correct Answer: A

```
package com.coderbd.Q17;


import java.beans.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;


/**
 *
 * @author Touhid
 */
public class Test {
   public static void main(String[] args)throws SQLException {
       //code to go here and regidter valid jdbc driver go here
       Connection con=DriverManager.getConnection(url, username, password);
       Statement
st=con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
       st.execute("SELECT * FROM student");
       ResultSet rs=st.getResultSet();
       rs.absolute(3);
       rs.moveToInsertRow();
       rs.updateInt(1, 113);
       rs.updateString(2, "Jannet");
       rs.updateString(3, "jannet@uni.com");
       rs.insertRow();
       System.out.println(rs.getInt(1)+" : "+rs.getString(2)+" : "+rs.getString(3));
   }
```

}

Problem 18:



Which two are elements of a singleton class?

□ A) a `public` method to instantiate the single instance

□ B) a `private static` reference to point to the single instance

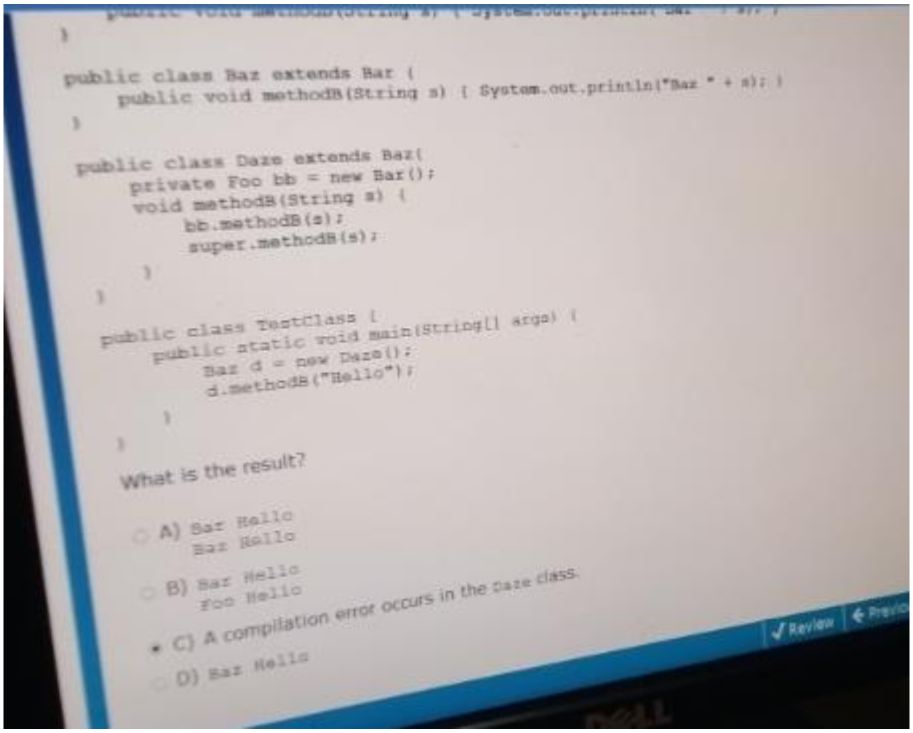
□ C) a `public` constructor to the class

□ D) a `transient` reference to point to the single instance

□ E) a `public static` method to return a copy of the singleton reference

Correct Answer: B, E

Problem 19:



```
    public void methodB(String s) { System.out.println("Bar " + s); }
}
public class Baz extends Bar {
    public void methodB(String s) { System.out.println("Baz " + s); }
}
public class Daze extends Baz{
    private Foo bb = new Bar();
    void methodB(String s) {
        bb.methodB(s);
        super.methodB(s);
    }
}
public class TestClass {
    public static void main(String[] args) {
        Baz d = new Daze();
        d.methodB("Hello");
    }
}
```

What is the result?

○ A) `Bar Hello`
`Baz Hello`

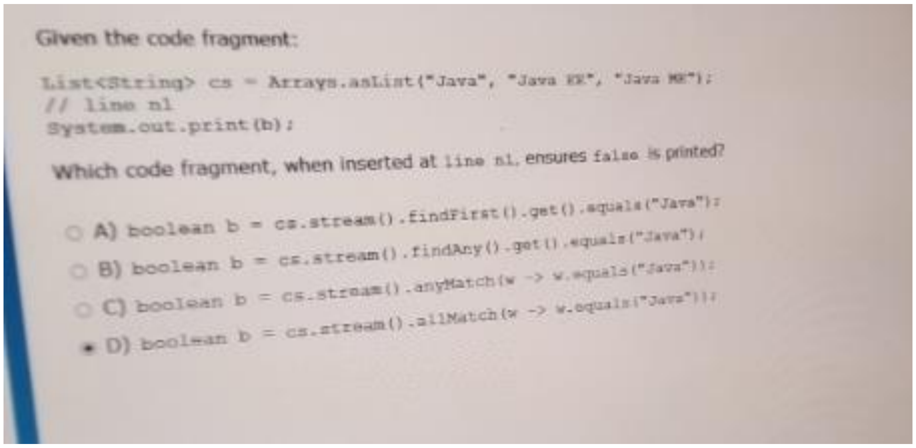
○ B) `Bar Hello`
`Foo Hello`

● C) A compilation error occurs in the `Daze` class.

○ D) `Baz Hello`

Correct Answer:

Problem 20:



Given the code fragment:

```
List<String> cs = Arrays.asList("Java", "Java EE", "Java ME");
// line n1
System.out.print(b);
```

Which code fragment, when inserted at `line n1`, ensures `false` is printed?

○ A) `boolean b = cs.stream().findFirst().get().equals("Java");`

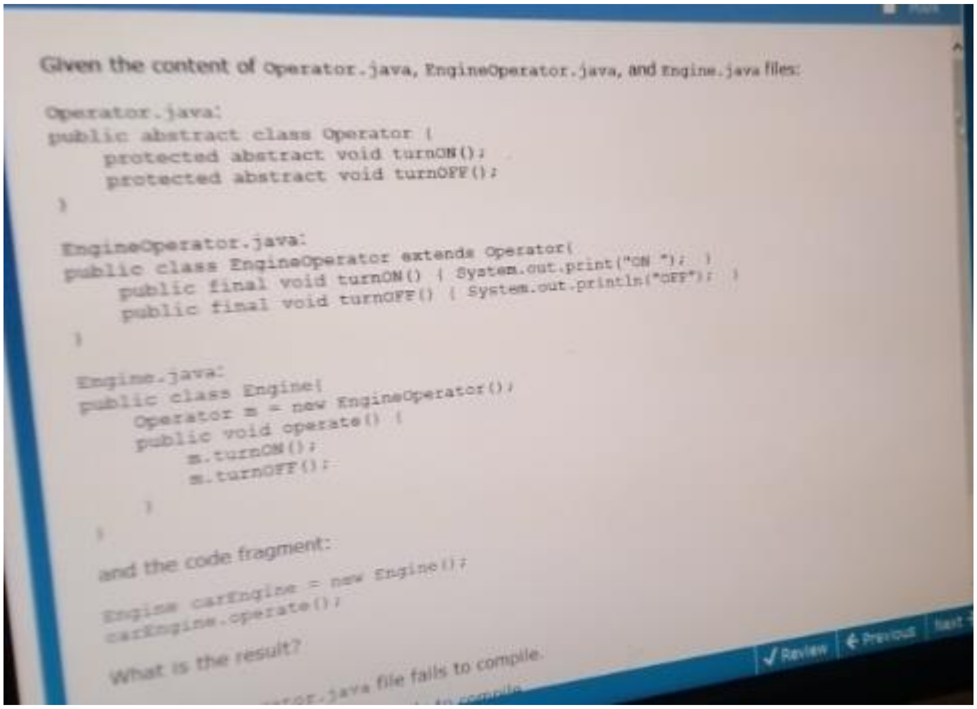
○ B) `boolean b = cs.stream().findAny().get().equals("Java");`

○ C) `boolean b = cs.stream().anyMatch(w -> w.equals("Java"));`

● D) `boolean b = cs.stream().allMatch(w -> w.equals("Java"));`

Correct Answer: D

Problem 21:



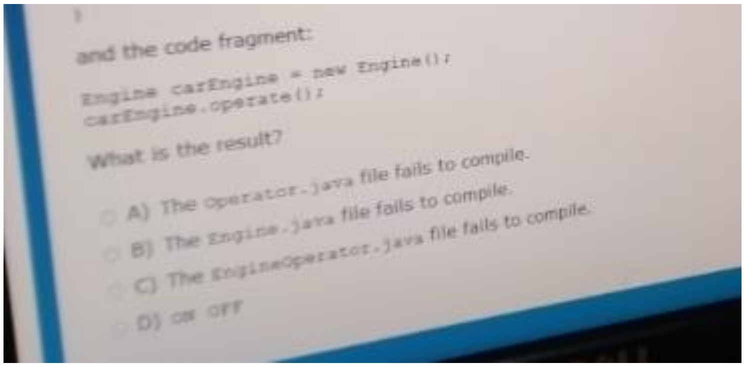
Given the content of `Operator.java`, `EngineOperator.java`, and `Engine.java` files:

```
Operator.java:
public abstract class Operator {
    protected abstract void turnON();
    protected abstract void turnOFF();
}

EngineOperator.java:
public class EngineOperator extends Operator{
    public final void turnON() { System.out.print("ON "); }
    public final void turnOFF() { System.out.println("OFF"); }
}

Engine.java:
public class Engine{
    Operator m = new EngineOperator();
    public void operate() {
        m.turnON();
        m.turnOFF();
    }
}
```

and the code fragment:

```
Engine carEngine = new Engine();
carEngine.operate();
```

What is the result?



and the code fragment:

```
Engine carEngine = new Engine();
carEngine.operate();
```

What is the result?

○ A) The `Operator.java` file fails to compile.

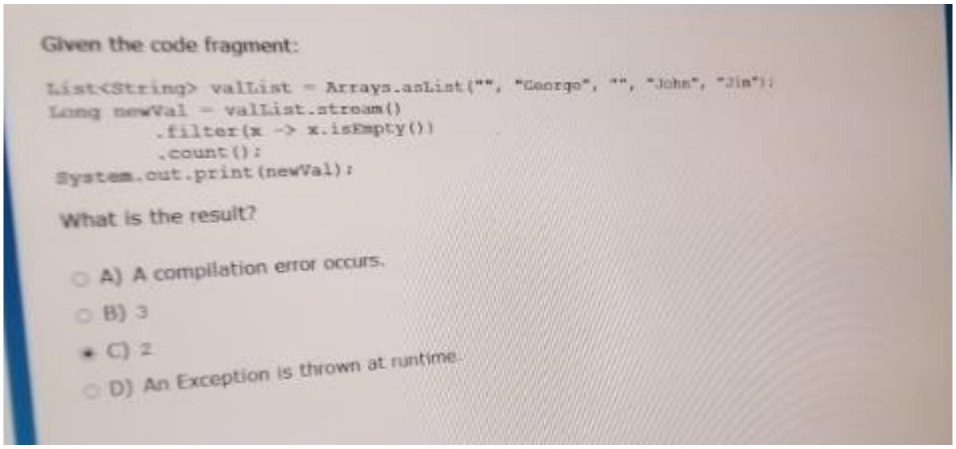
○ B) The `Engine.java` file fails to compile.

○ C) The `EngineOperator.java` file fails to compile.

○ D) `ON OFF`

Correct Answer:

Problem 22:



Given the code fragment:

```
List<String> valList = Arrays.asList("", "George", "", "John", "Jim");
Long newVal = valList.stream()
            .filter(x -> x.isEmpty())
            .count();
System.out.print(newVal);
```

What is the result?

○ A) A compilation error occurs.

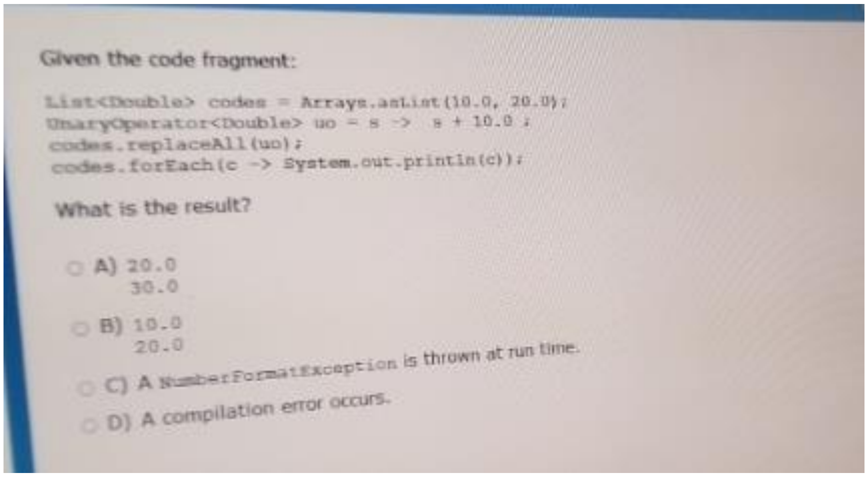
○ B) 3

● C) 2

○ D) An Exception is thrown at runtime.

Correct Answer: C

Problem 23:



Given the code fragment:

```
List<Double> codes = Arrays.asList(10.0, 20.0);
UnaryOperator<Double> uo = s ->  s + 10.0 ;
codes.replaceAll(uo);
codes.forEach(c -> System.out.println(c));
```

What is the result?

○ A) 20.0
30.0

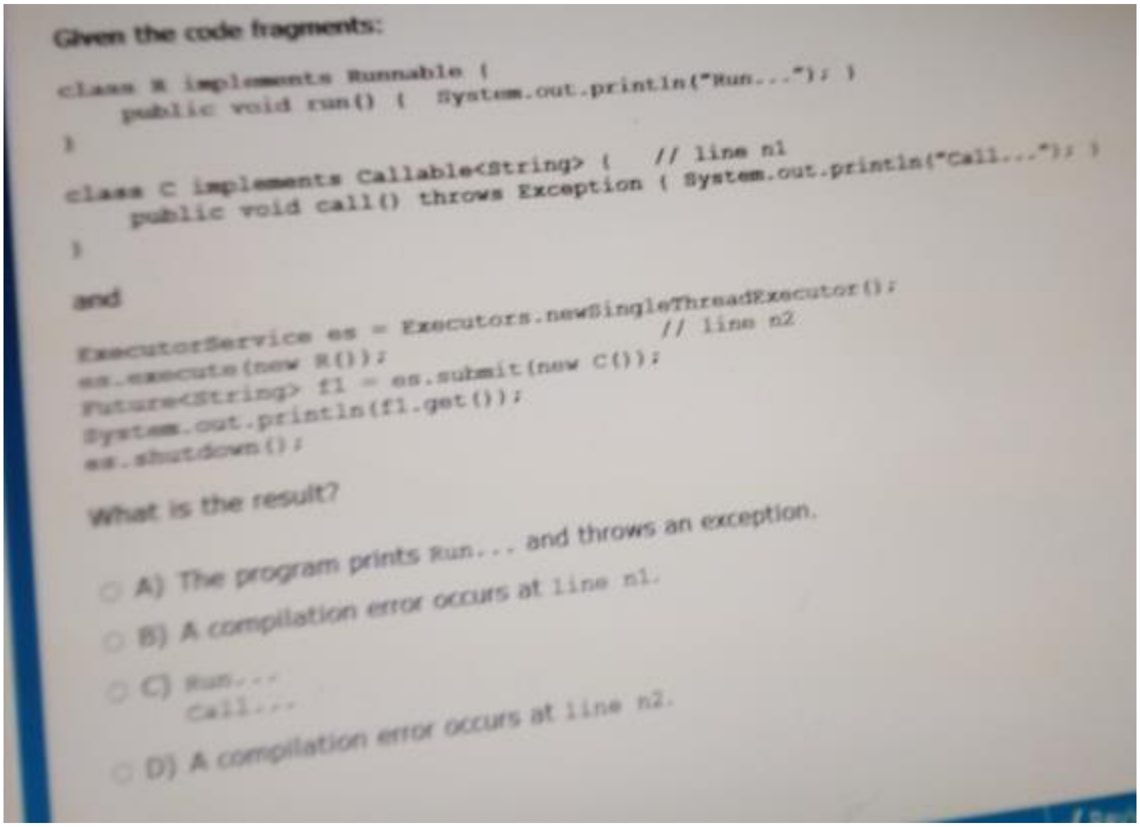
○ B) 10.0
20.0

○ C) A NumberFormatException is thrown at run time.

○ D) A compilation error occurs.

Correct Answer:

Problem 24:



Given the code fragments:

```
class R implements Runnable {
    public void run() {  System.out.println("Run..."); }
}
class C implements Callable<String> {    // line n1
    public void call() throws Exception { System.out.println("Call..."); }
}
```

and

```
ExecutorService es = Executors.newSingleThreadExecutor();
                                             // line n2
es.execute(new R());
Future<String> f1 = es.submit(new C());
System.out.println(f1.get());
es.shutdown();
```

What is the result?

○ A) The program prints Run... and throws an exception.

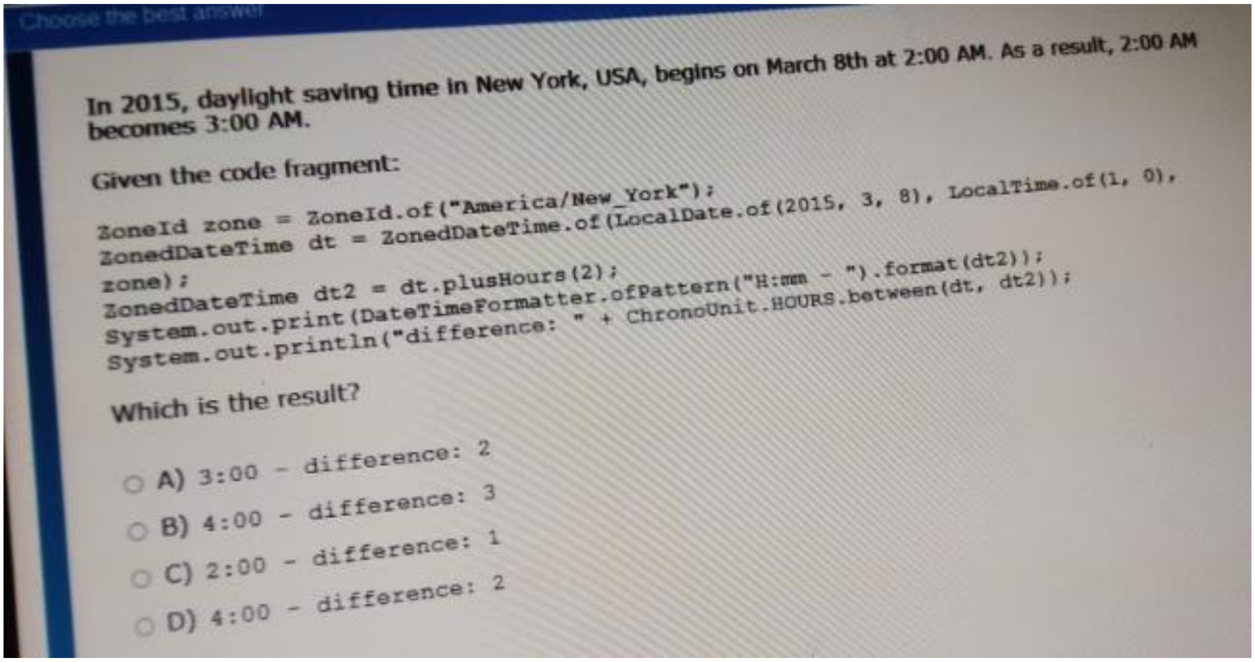
○ B) A compilation error occurs at line n1.

○ C) Run...
Call...

○ D) A compilation error occurs at line n2.

Correct Answer:

Problem 25:



In 2015, daylight saving time in New York, USA, begins on March 8th at 2:00 AM. As a result, 2:00 AM becomes 3:00 AM.

Given the code fragment:

```
ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime dt = ZonedDateTime.of(LocalDate.of(2015, 3, 8), LocalTime.of(1, 0),
zone);
ZonedDateTime dt2 = dt.plusHours(2);
System.out.print(DateTimeFormatter.ofPattern("H:mm - ").format(dt2));
System.out.println("difference: " + ChronoUnit.HOURS.between(dt, dt2));
```

Which is the result?

○ A) `3:00 - difference: 2`

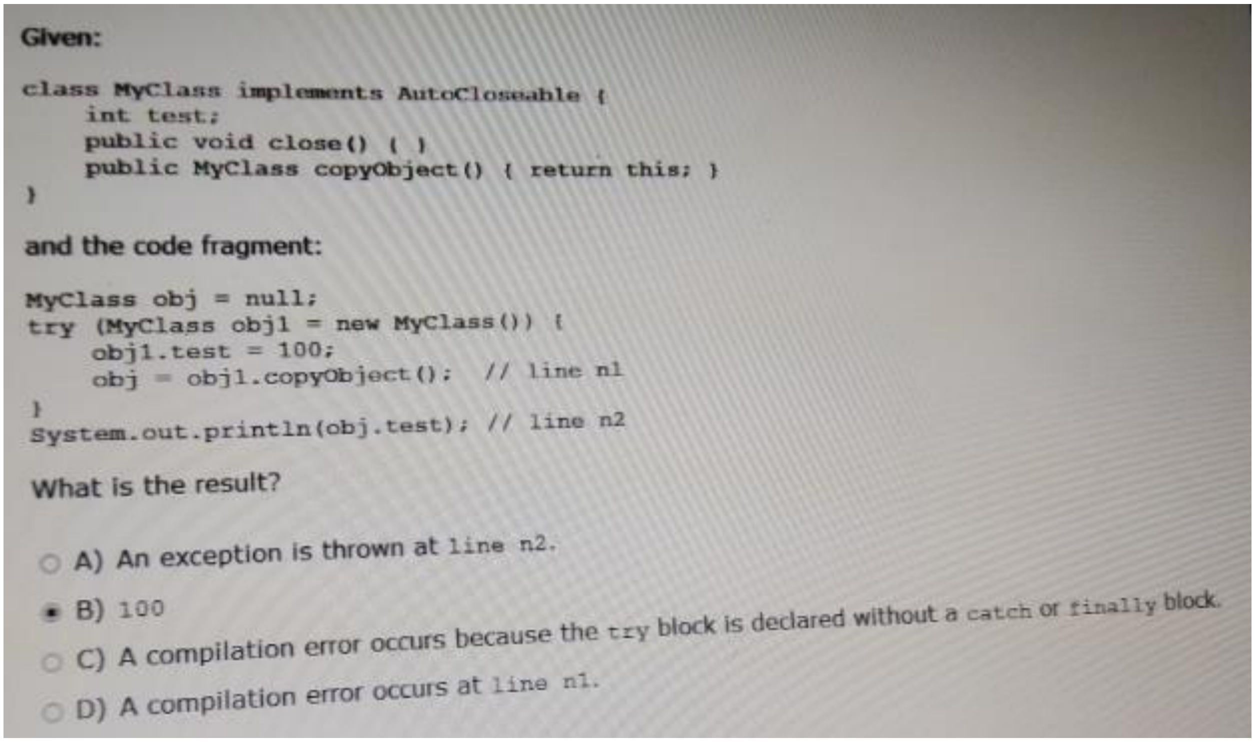
○ B) `4:00 - difference: 3`

○ C) `2:00 - difference: 1`

○ D) `4:00 - difference: 2`

Correct Answer:

Problem 26:



Given:

```
class MyClass implements AutoCloseable {
    int test;
    public void close() { }
    public MyClass copyObject() { return this; }
}
```

and the code fragment:

```
MyClass obj = null;
try (MyClass obj1 = new MyClass()) {
    obj1.test = 100;
    obj = obj1.copyObject();  // line n1
}
System.out.println(obj.test); // line n2
```

What is the result?

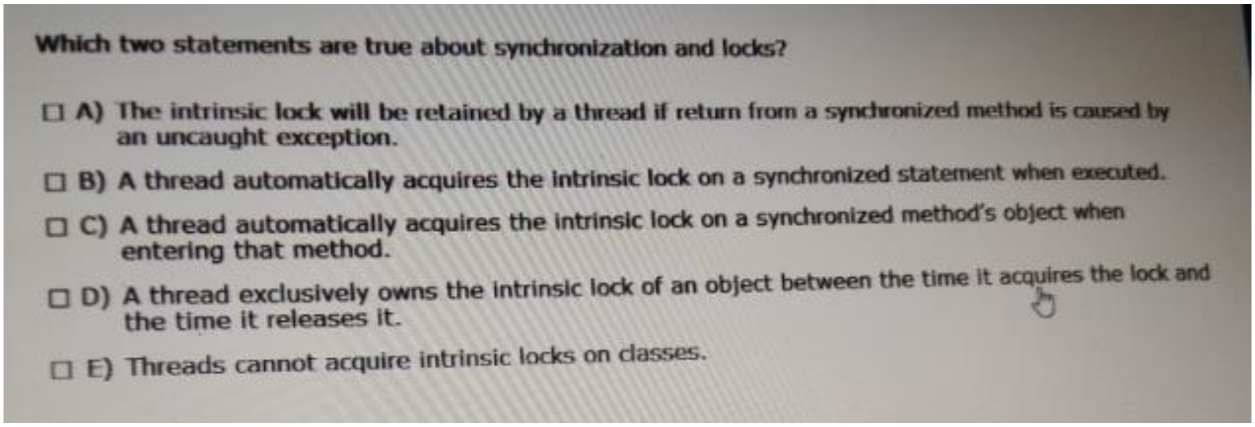
○ A) An exception is thrown at `line n2`.

● B) `100`

○ C) A compilation error occurs because the `try` block is declared without a `catch` or `finally` block.

○ D) A compilation error occurs at `line n1`.

Problem 27:



Which two statements are true about synchronization and locks?

□ A) The intrinsic lock will be retained by a thread if return from a synchronized method is caused by an uncaught exception.

□ B) A thread automatically acquires the intrinsic lock on a synchronized statement when executed.

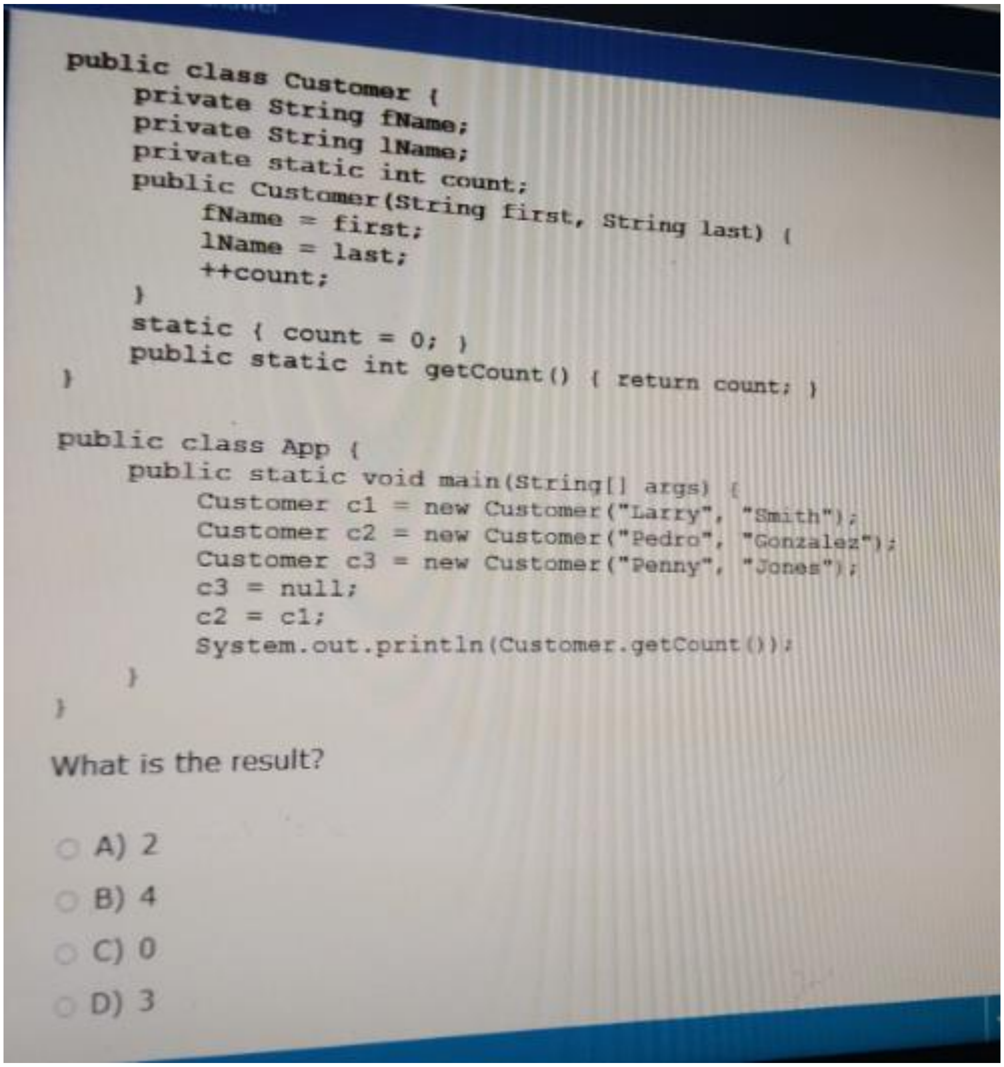
□ C) A thread automatically acquires the intrinsic lock on a synchronized method's object when entering that method.

□ D) A thread exclusively owns the intrinsic lock of an object between the time it acquires the lock and the time it releases it.

□ E) Threads cannot acquire intrinsic locks on classes.

Correct Answer: A, B

Problem 28:



```
public class Customer {
    private String fName;
    private String lName;
    private static int count;
    public Customer(String first, String last) {
        fName = first;
        lName = last;
        ++count;
    }
    static { count = 0; }
    public static int getCount() { return count; }
}

public class App {
    public static void main(String[] args) {
        Customer c1 = new Customer("Larry", "Smith");
        Customer c2 = new Customer("Pedro", "Gonzalez");
        Customer c3 = new Customer("Penny", "Jones");
        c3 = null;
        c2 = c1;
        System.out.println(Customer.getCount());
    }
}
```

What is the result?

○ A) 2

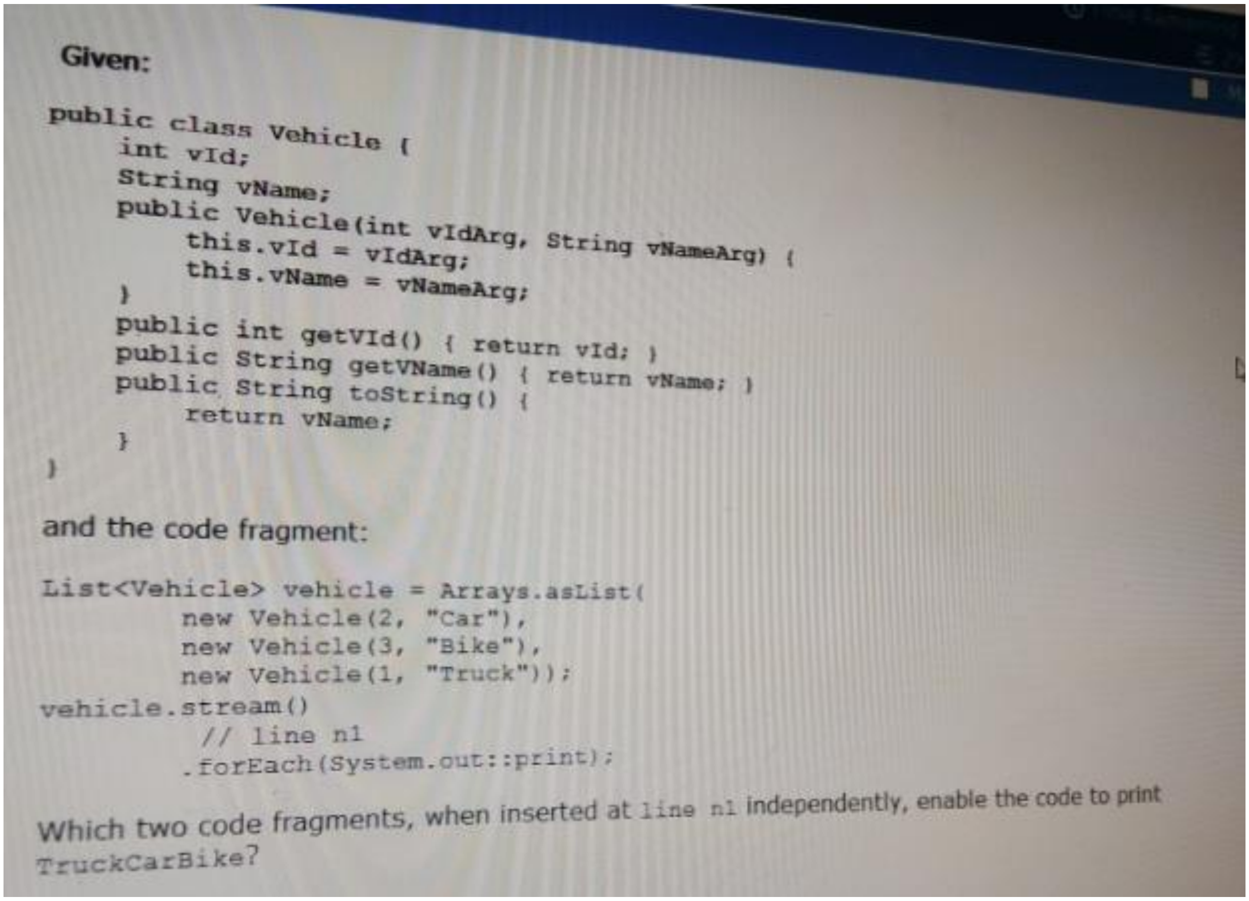
○ B) 4

○ C) 0

○ D) 3

Correct Answer:

Problem 29:



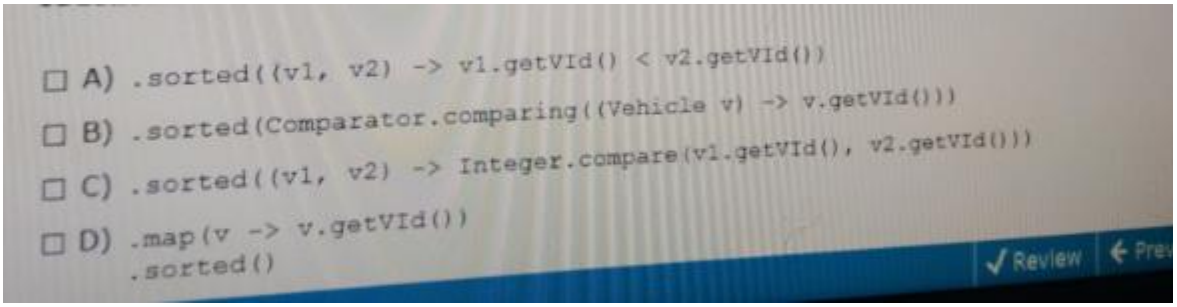
Given:

```
public class Vehicle {
     int vId;
     String vName;
     public Vehicle(int vIdArg, String vNameArg) {
          this.vId = vIdArg;
          this.vName = vNameArg;
     }
     public int getVId() { return vId; }
     public String getVName() { return vName; }
     public String toString() {
          return vName;
     }
}
```

and the code fragment:

```
List<Vehicle> vehicle = Arrays.asList(
          new Vehicle(2, "Car"),
          new Vehicle(3, "Bike"),
          new Vehicle(1, "Truck"));
vehicle.stream()
           // line n1
          .forEach(System.out::print);
```

Which two code fragments, when inserted at `line n1` independently, enable the code to print `TruckCarBike`?



□ A) `.sorted((v1, v2) -> v1.getVId() < v2.getVId())`

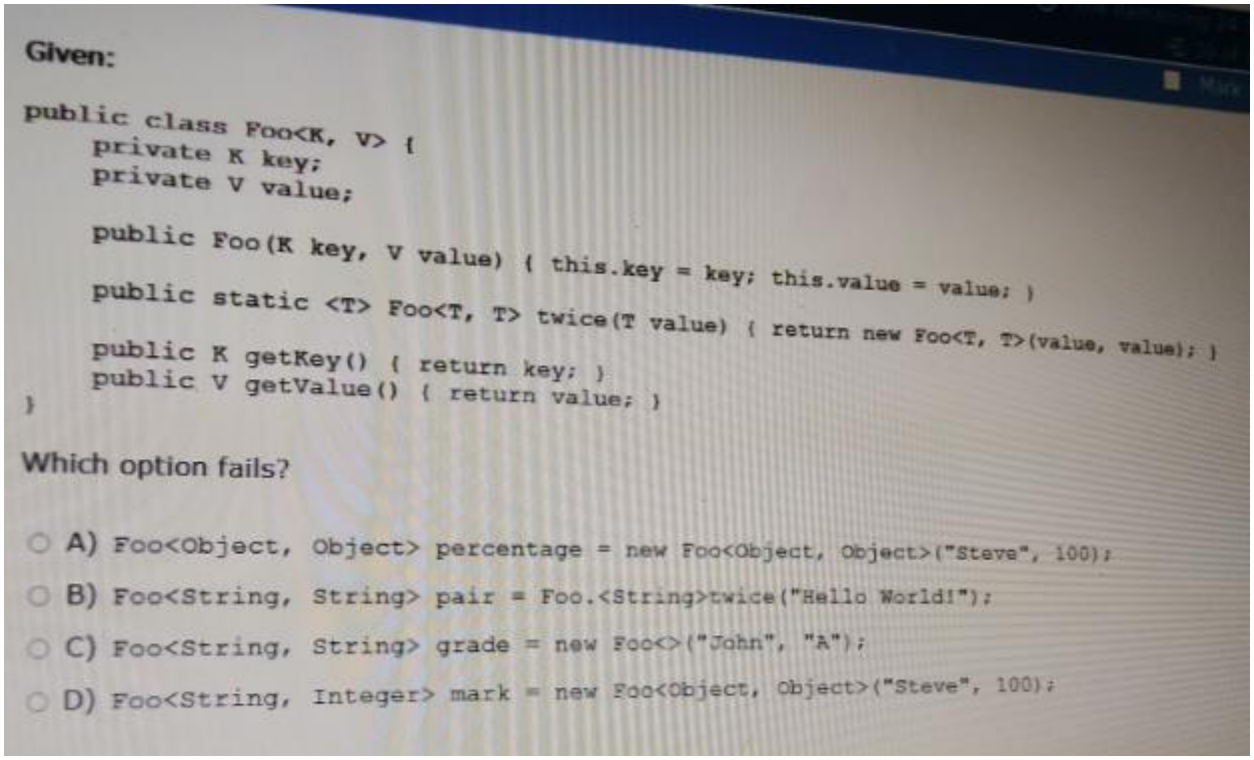
□ B) `.sorted(Comparator.comparing((Vehicle v) -> v.getVId()))`

□ C) `.sorted((v1, v2) -> Integer.compare(v1.getVId(), v2.getVId()))`

□ D) `.map(v -> v.getVId())`
`.sorted()`

Correct Answer:

Problem 30:



Given:

```
public class Foo<K, V> {
    private K key;
    private V value;

    public Foo(K key, V value) { this.key = key; this.value = value; }

    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }

    public K getKey() { return key; }
    public V getValue() { return value; }
}
```

Which option fails?

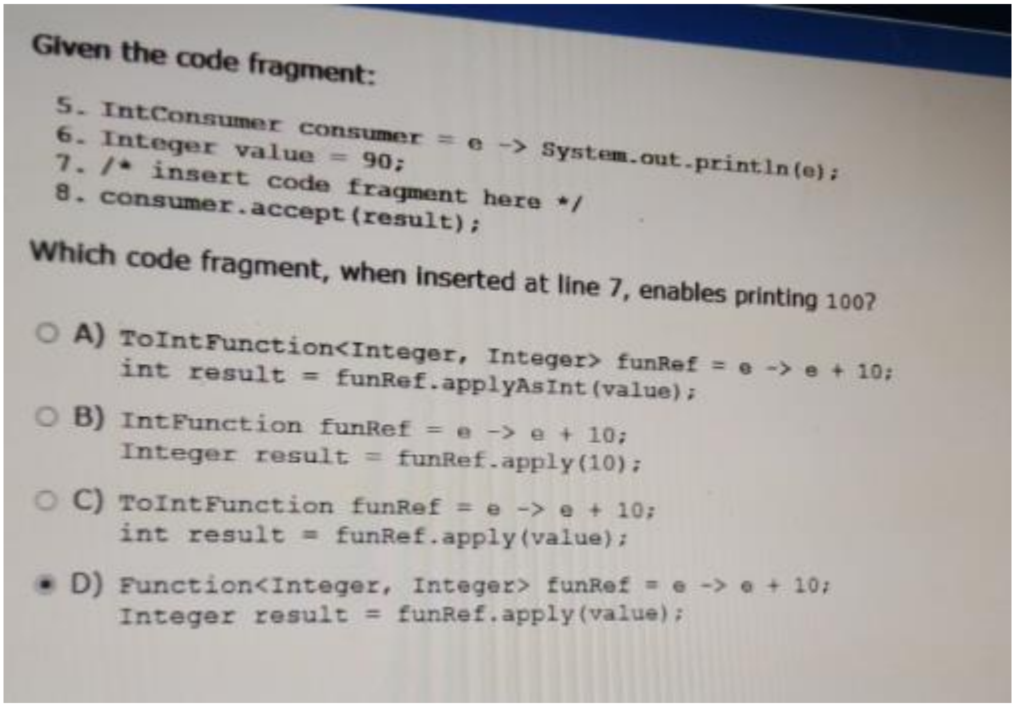
○ A) `Foo<Object, Object> percentage = new Foo<Object, Object>("Steve", 100);`

○ B) `Foo<String, String> pair = Foo.<String>twice("Hello World!");`

○ C) `Foo<String, String> grade = new Foo<>("John", "A");`

○ D) `Foo<String, Integer> mark = new Foo<Object, Object>("Steve", 100);`

Problem 31:



Given the code fragment:

```
5. IntConsumer consumer = e -> System.out.println(e);
6. Integer value = 90;
7. /* insert code fragment here */
8. consumer.accept(result);
```

Which code fragment, when inserted at line 7, enables printing 100?

○ A)
```
ToIntFunction<Integer, Integer> funRef = e -> e + 10;
int result = funRef.applyAsInt(value);
```

○ B)
```
IntFunction funRef = e -> e + 10;
Integer result = funRef.apply(10);
```

○ C)
```
ToIntFunction funRef = e -> e + 10;
int result = funRef.apply(value);
```

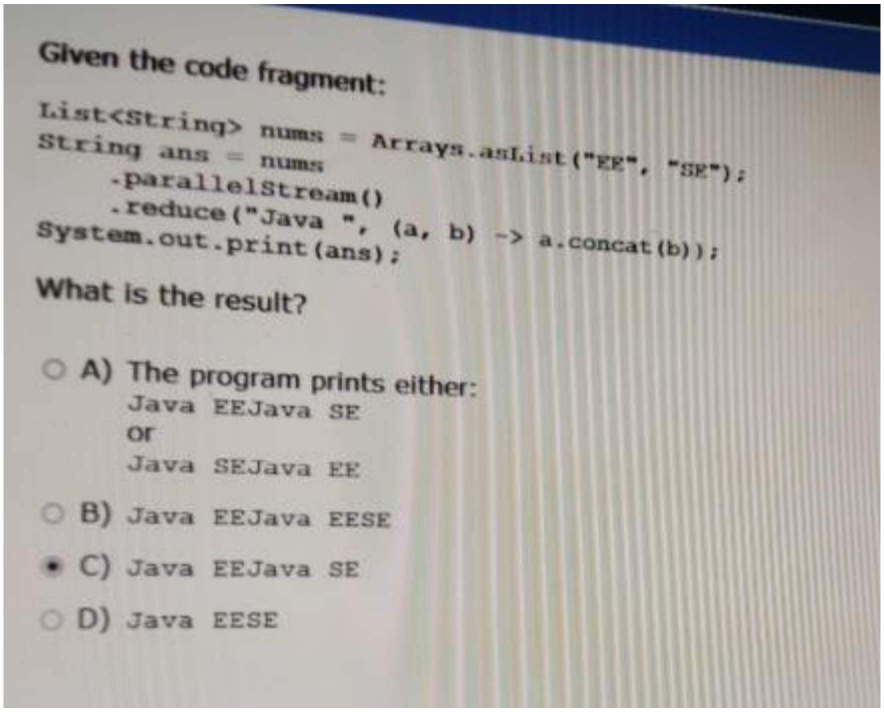
● D)
```
Function<Integer, Integer> funRef = e -> e + 10;
Integer result = funRef.apply(value);
```

Correct Answer: D

package com.coderbd.q31;

```
import java.util.function.Function;

import java.util.function.IntConsumer;


/**

 * Answer: D

 */

public class Test {

    public static void main(String[] args) {

        IntConsumer consumer = e -> System.out.println(e);

        Integer value = 90;


        /* Inserted Code */

        Function<Integer, Integer> funRef = e -> e + 10;

        Integer result = funRef.apply(value);


        consumer.accept(result);

    }

}
```

Problem 32:



Given the code fragment:

```
List<String> nums = Arrays.asList("EE", "SE");
String ans = nums
     .parallelStream()
     .reduce("Java ", (a, b) -> a.concat(b));
System.out.print(ans);
```

What is the result?

○ A) The program prints either:
```
Java EEJava SE
```
or
```
Java SEJava EE
```

○ B) `Java EEJava EESE`

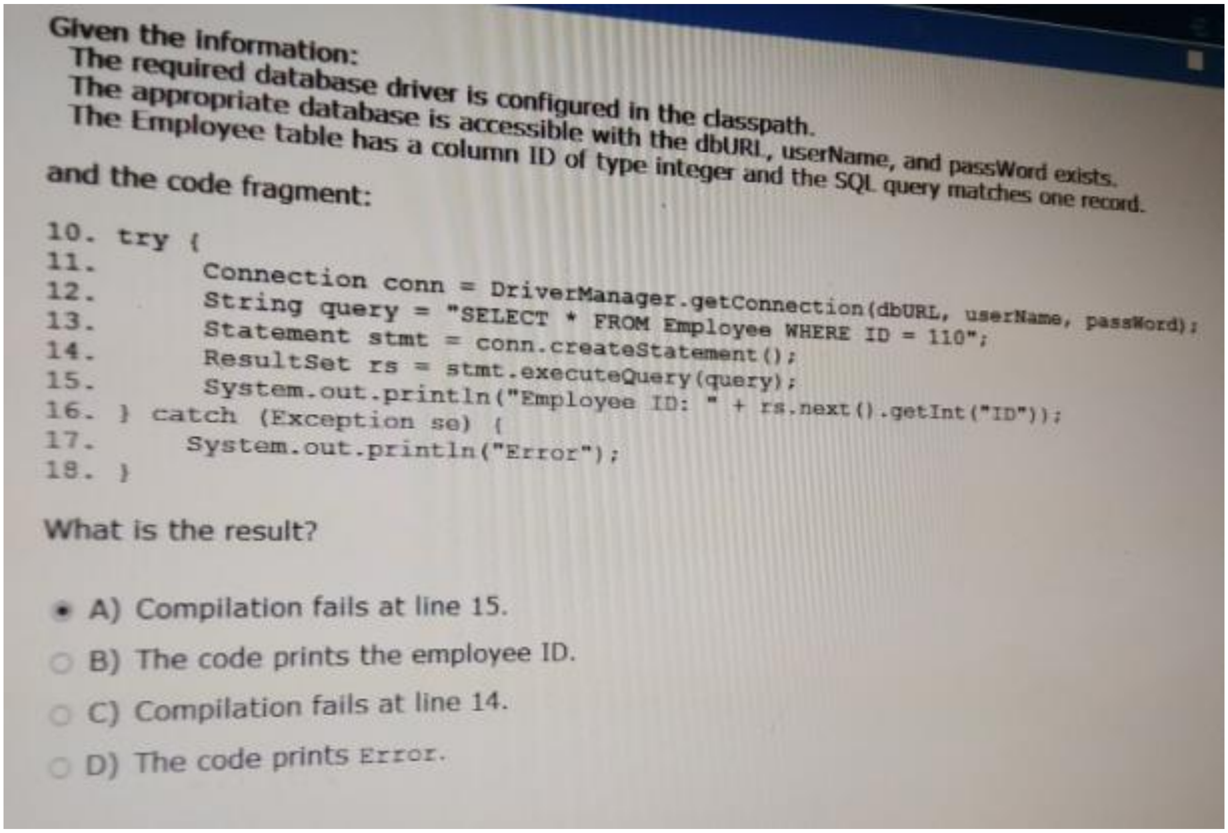
● C) `Java EEJava SE`

○ D) `Java EESE`

Correct Answer: C

```
package com.coderbd.q32;


import java.util.Arrays;

import java.util.List;


/**

 * Answer: C

 */

public class Test {

    public static void main(String[] args) {

        List<String> nums = Arrays.asList("EE", "SE");

        String ans = nums

                .parallelStream()
        .reduce("Java ", (a, b) -> a.concat(b));
    System.out.println(ans);
  }
}
```

Problem 33:



Given the information:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists.

The Employee table has a column ID of type integer and the SQL query matches one record.

and the code fragment:

```
10. try {
11.         Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
12.         String query = "SELECT * FROM Employee WHERE ID = 110";
13.         Statement stmt = conn.createStatement();
14.         ResultSet rs = stmt.executeQuery(query);
15.         System.out.println("Employee ID: " + rs.next().getInt("ID"));
16. } catch (Exception se) {
17.        System.out.println("Error");
18. }
```

What is the result?

● A) Compilation fails at line 15.

○ B) The code prints the employee ID.

○ C) Compilation fails at line 14.

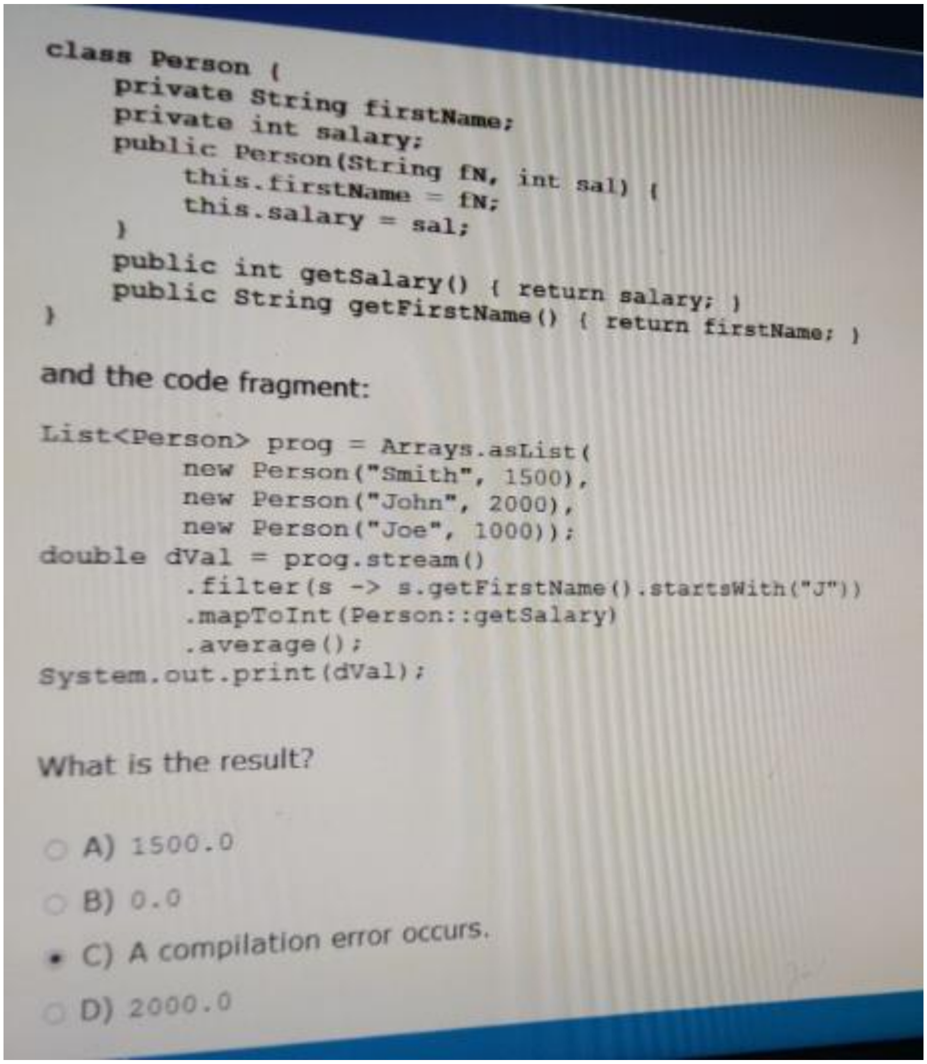
○ D) The code prints Error.

Correct Answer: A

```
package com.coderbd.q33;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

/**
```

```
 * Answer: A
 */
public class Test {
    public static void main(String[] args) {
        /* Given the information */
        String dbURL = "dbLink";
        String userName = "an username";
        String passWord = "password";

        try {
            Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
            String query = "SELECT * FROM Employee WHERE ID = 110";
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            System.out.println("Employee ID: "+ rs.next().getInt("ID"));
        } catch (Exception se) {
            System.out.println("Error");
        }
    }
}
```

Problem 34:



```
class Person {
    private String firstName;
    private int salary;
    public Person(String fN, int sal) {
        this.firstName = fN;
        this.salary = sal;
    }
    public int getSalary() { return salary; }
    public String getFirstName() { return firstName; }
}
```

and the code fragment:

```
List<Person> prog = Arrays.asList(
        new Person("Smith", 1500),
        new Person("John", 2000),
        new Person("Joe", 1000));
double dVal = prog.stream()
        .filter(s -> s.getFirstName().startsWith("J"))
        .mapToInt(Person::getSalary)
        .average();
System.out.print(dVal);
```

What is the result?

○ A) 1500.0

○ B) 0.0

● C) A compilation error occurs.

○ D) 2000.0

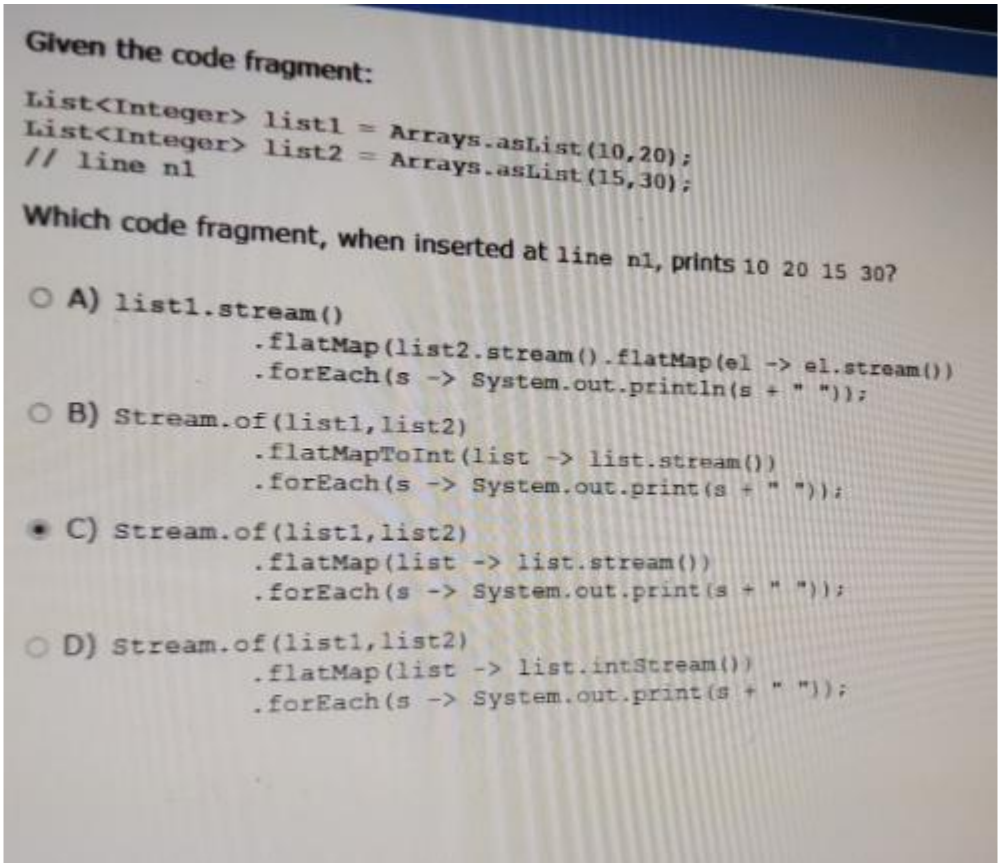
Correct Answer: C

package com.coderbd.q34;

import java.util.Arrays;

```
import java.util.List;

/**
 * Answer: C
 */
class Person {
    private String firstName;
    private int salary;

    public Person(String fN, int sal) {
        this.firstName = fN;
        this.salary = sal;
    }

    public int getSalary() {
        return salary;
    }

    public String getFirstName() {
        return firstName;
    }
}

public class Test {
    public static void main(String[] args) {
        List<Person> prog = Arrays.asList(
                new Person("Smith", 1500),
                new Person("John", 2000),
                new Person("Joe", 1000));
        double dVal = prog.stream()
                .filter(s -> s.getFirstName().startsWith("J"))
                .mapToInt(Person::getSalary)
                .average();
        System.out.println(dVal);
    }
}
```

Problem 35:



Given the code fragment:

```
List<Integer> list1 = Arrays.asList(10,20);
List<Integer> list2 = Arrays.asList(15,30);
// line n1
```

Which code fragment, when inserted at line n1, prints 10 20 15 30?

○ A)
```
list1.stream()
        .flatMap(list2.stream().flatMap(el -> el.stream())
        .forEach(s -> System.out.println(s + " "));
```

○ B)
```
Stream.of(list1,list2)
        .flatMapToInt(list -> list.stream())
        .forEach(s -> System.out.print(s + " "));
```

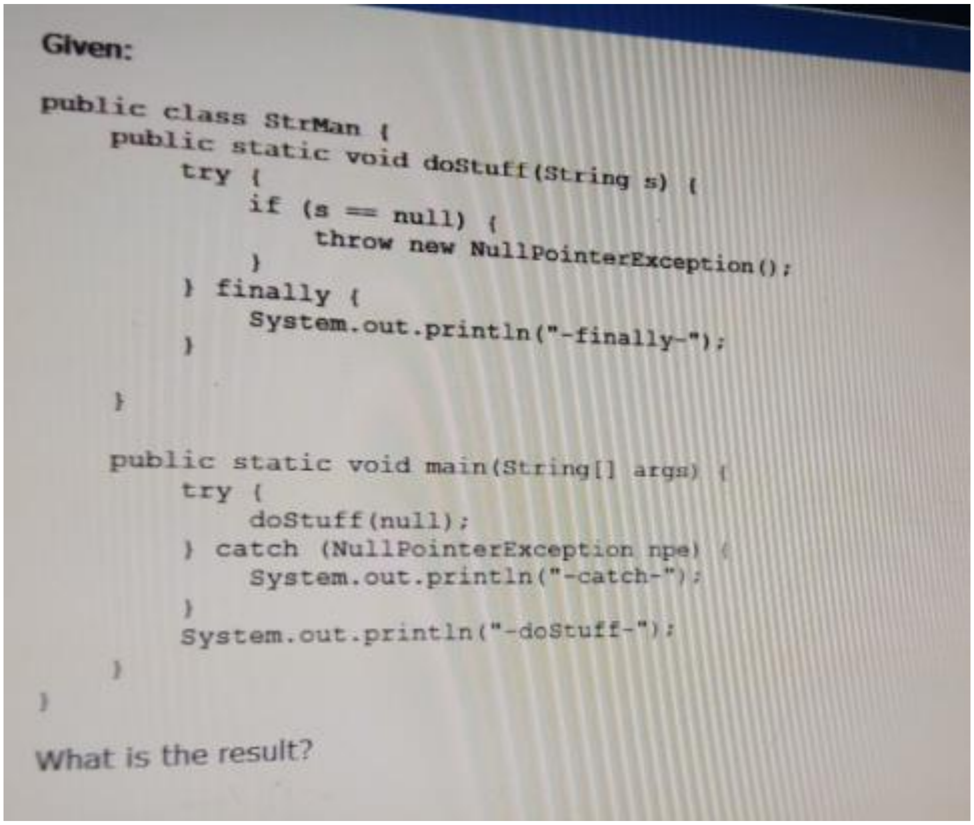
● C)
```
Stream.of(list1,list2)
        .flatMap(list -> list.stream())
        .forEach(s -> System.out.print(s + " "));
```

○ D)
```
Stream.of(list1,list2)
        .flatMap(list -> list.intStream())
        .forEach(s -> System.out.print(s + " "));
```

Correct Answer: C

```
package com.coderbd.q35;


import java.util.Arrays;

import java.util.List;
import java.util.stream.Stream;


/**
 * Answer: C
 */
public class Test {
    public static void main(String[] args) {
        List<Integer> list1 = Arrays.asList(10, 20);
        List<Integer> list2 = Arrays.asList(15, 30);


        // line n1
        Stream.of(list1, list2)
                .flatMap(list -> list.stream())
                .forEach(s -> System.out.println(s + " "));
    }
}
```

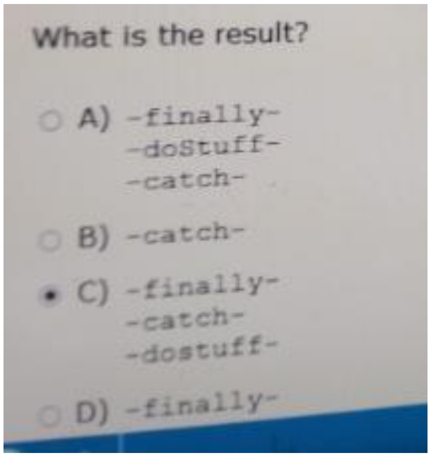
Problem 36:



Given:

```
public class StrMan {
    public static void doStuff(String s) {
        try {
            if (s == null) {
                throw new NullPointerException();
            }
        } finally {
            System.out.println("-finally-");
        }

    }

    public static void main(String[] args) {
        try {
            doStuff(null);
        } catch (NullPointerException npe) {
            System.out.println("-catch-");
        }
        System.out.println("-doStuff-");
    }
}
```

What is the result?



What is the result?

○ A) -finally-
-doStuff-
-catch-

○ B) -catch-

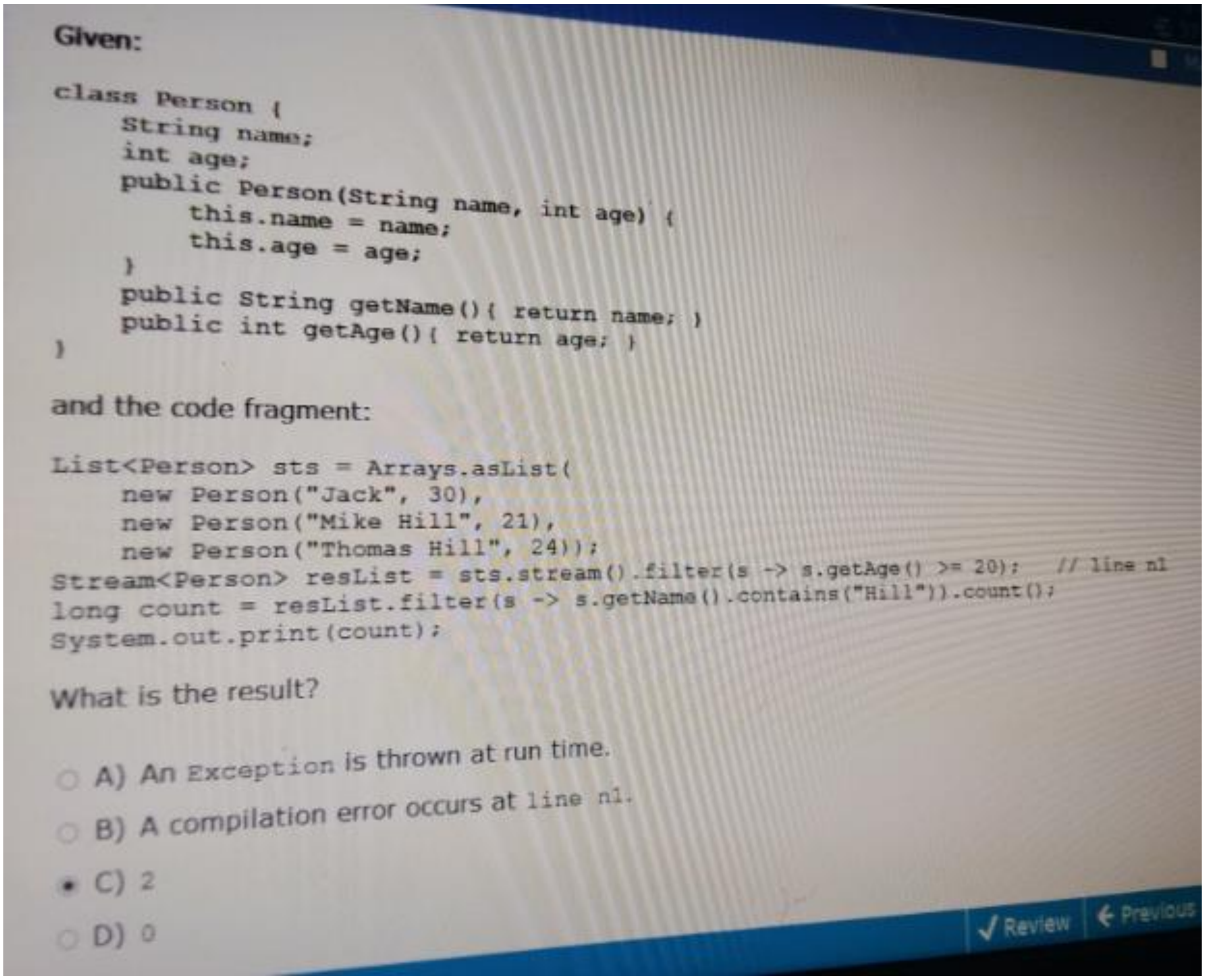
● C) -finally-
-catch-
-dostuff-

○ D) -finally-

Correct Answer: C

```
package com.coderbd.q36;

/**
 * Answer: C
 */
public class StrMan {
   public static void doStuff(String s) {
      try {
         if (s == null) {
            throw new NullPointerException();
         }
      } finally {
         System.out.println("-finally-");
      }
   }

   public static void main(String[] args) {
      try {
            doStuff(null);
        } catch (NullPointerException npe) {
            System.out.println("-catch-");
        }
        System.out.println("-doStuff-");
    }
}
```

Problem 37:



Given:

```
class Person {
    String name;
    int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName(){ return name; }
    public int getAge(){ return age; }
}
```

and the code fragment:

```
List<Person> sts = Arrays.asList(
    new Person("Jack", 30),
    new Person("Mike Hill", 21),
    new Person("Thomas Hill", 24));
Stream<Person> resList = sts.stream().filter(s -> s.getAge() >= 20);   // line n1
long count = resList.filter(s -> s.getName().contains("Hill")).count();
System.out.print(count);
```

What is the result?

○ A) An Exception is thrown at run time.

○ B) A compilation error occurs at line n1.

● C) 2

○ D) 0

Correct Answer: C

package com.coderbd.q37;

import java.util.Arrays;

import java.util.List;

import java.util.stream.Stream;

/**

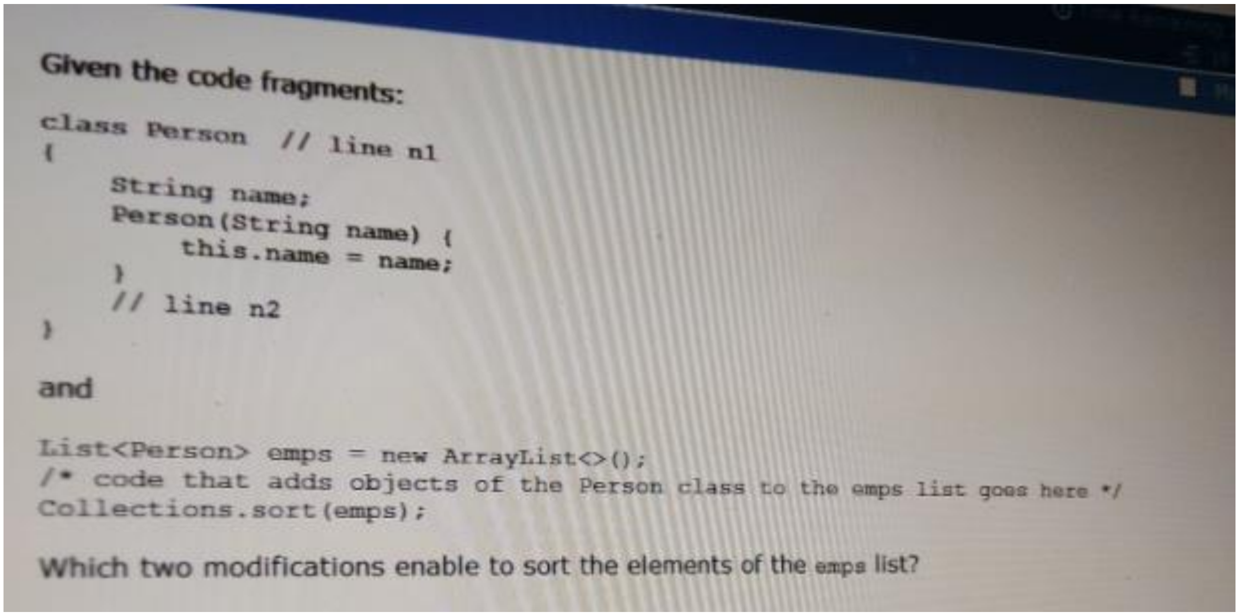
 * Answer: C

 */

class Person {

```
    String name;
    int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}

public class Test {
    public static void main(String[] args) {
        List<Person> sts = Arrays.asList(
                new Person("Jack", 30),
                new Person("Mike Hill", 21),
                new Person("Thomas Hill", 24));
        Stream<Person> resList = sts.stream().filter(s -> s.getAge() >= 20);
        long count = resList.filter(s -> s.getName().contains("Hill")).count();
        System.out.print(count);
    }
}
```

Problem 38:



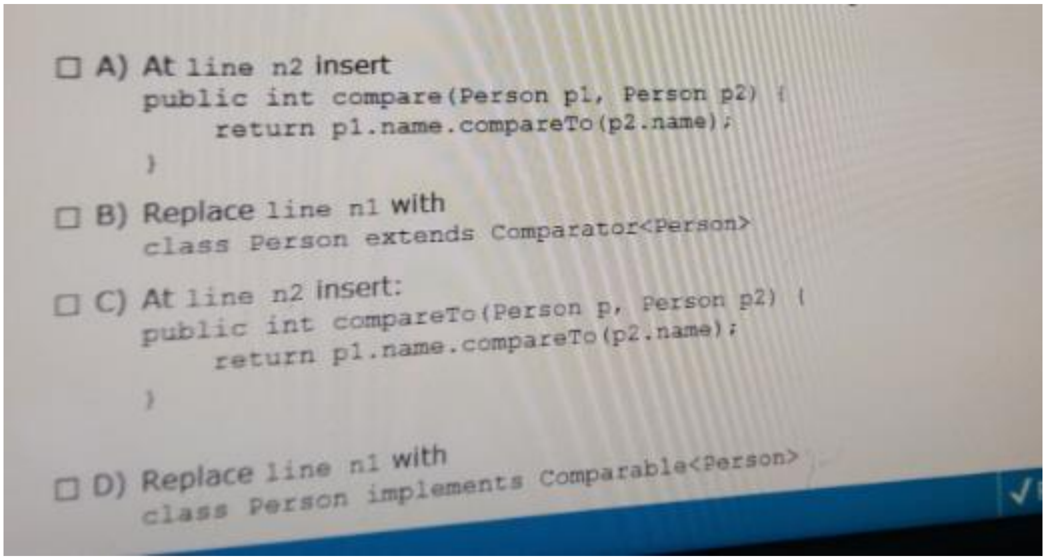
Given the code fragments:

```
class Person  // line n1
{
     String name;
     Person(String name) {
          this.name = name;
     }
     // line n2
}
```

and

```
List<Person> emps = new ArrayList<>();
/* code that adds objects of the Person class to the emps list goes here */
Collections.sort(emps);
```

Which two modifications enable to sort the elements of the emps list?



☐ A) At line n2 insert

```
public int compare(Person p1, Person p2) {
     return p1.name.compareTo(p2.name);
}
```

☐ B) Replace line n1 with

```
class Person extends Comparator<Person>
```

☐ C) At line n2 insert:

```
public int compareTo(Person p, Person p2) {
     return p1.name.compareTo(p2.name);
}
```

☐ D) Replace line n1 with

```
class Person implements Comparable<Person>
```

Correct Answer: E, F

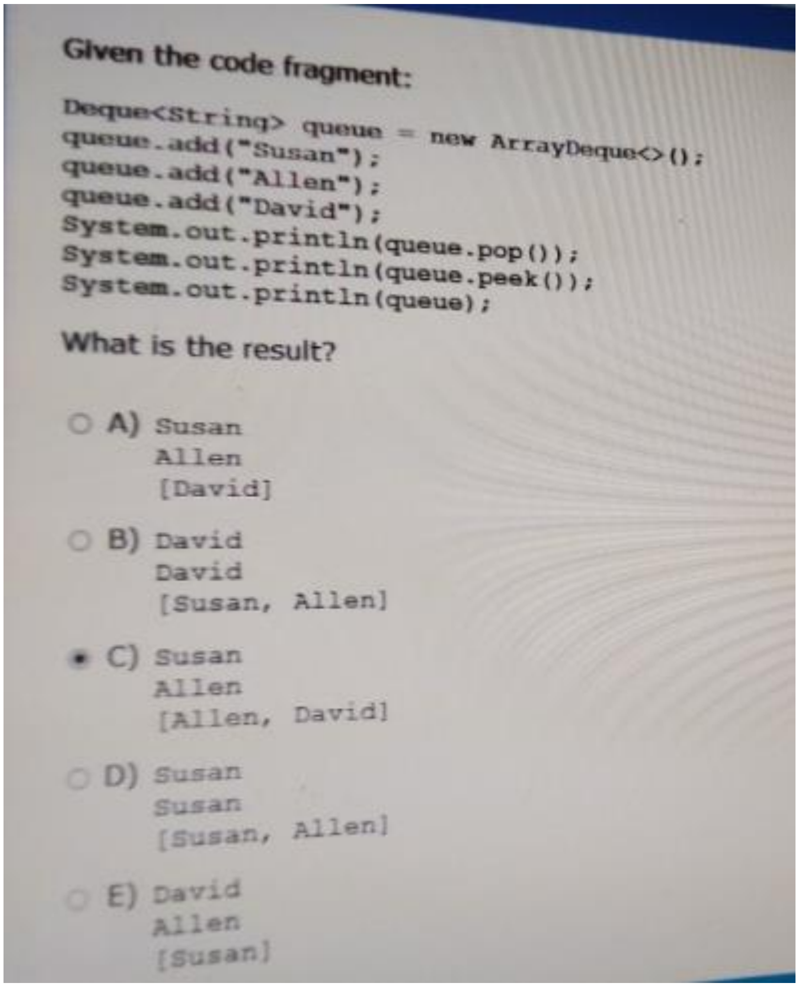
package com.coderbd.q38;

import java.util.*;

/**

```
 * Answer: E, F
 */
class Person implements Comparable<Person> // line n1 @ E
{
    String name;

    Person(String name) {
        this.name = name;
    }

    // line n2 @ F
    public int compareTo(Person p) {
        return this.name.compareTo(p.name);
    }
}

public class Test {
    public static void main(String[] args) {
        List<Person> emps = new ArrayList<>();

        // Added some values to emps for testing purpose
        emps.add(new Person("Bappy"));
        emps.add(new Person("Zahid"));
        emps.add(new Person("Xian"));
        emps.add(new Person("Asad"));

        Collections.sort(emps);

        // Added for testing purpose
        System.out.println(emps.get(0).name);
    }
}
```

Problem 39:



Given the code fragment:

```
Deque<String> queue = new ArrayDeque<>();
queue.add("Susan");
queue.add("Allen");
queue.add("David");
System.out.println(queue.pop());
System.out.println(queue.peek());
System.out.println(queue);
```

What is the result?

○ A)
```
Susan
Allen
[David]
```

○ B)
```
David
David
[Susan, Allen]
```

● C)
```
Susan
Allen
[Allen, David]
```

○ D)
```
Susan
Susan
[Susan, Allen]
```

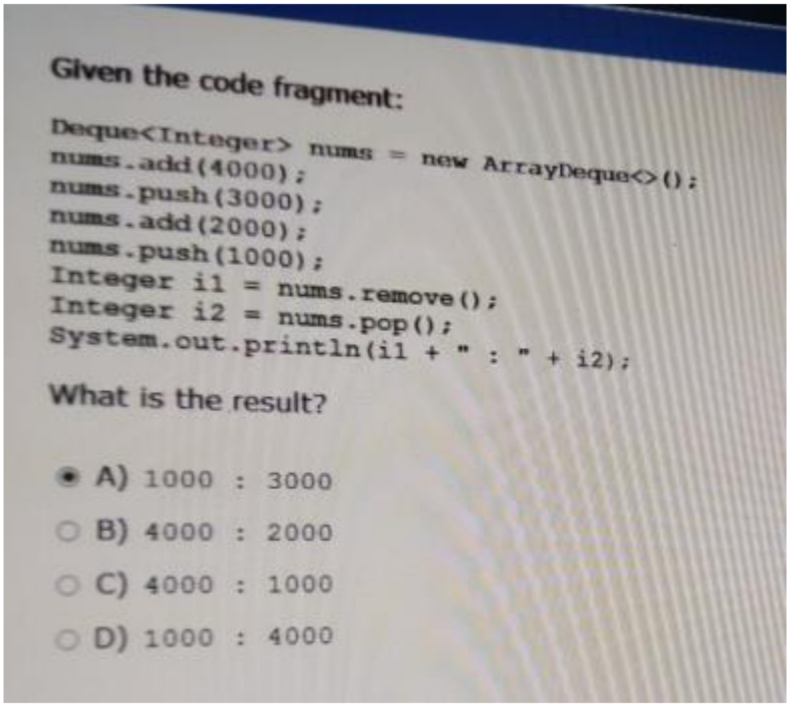
○ E)
```
David
Allen
[Susan]
```

Correct Answer: C

```
package com.coderbd.q39;

import java.util.ArrayDeque;
import java.util.Deque;

/**
```

```
 * Answer: C
 */
public class Test {
   public static void main(String[] args) {
      Deque<String> queue = new ArrayDeque<>();
      queue.add("Susan");
      queue.add("Allen");
      queue.add("David");
      System.out.println(queue.pop());
      System.out.println(queue.peek());
      System.out.println(queue);
   }
}
```

Problem 40:



Given the code fragment:

```
Deque<Integer> nums = new ArrayDeque<>();
nums.add(4000);
nums.push(3000);
nums.add(2000);
nums.push(1000);
Integer i1 = nums.remove();
Integer i2 = nums.pop();
System.out.println(i1 + " : " + i2);
```

What is the result?

● A) 1000 : 3000

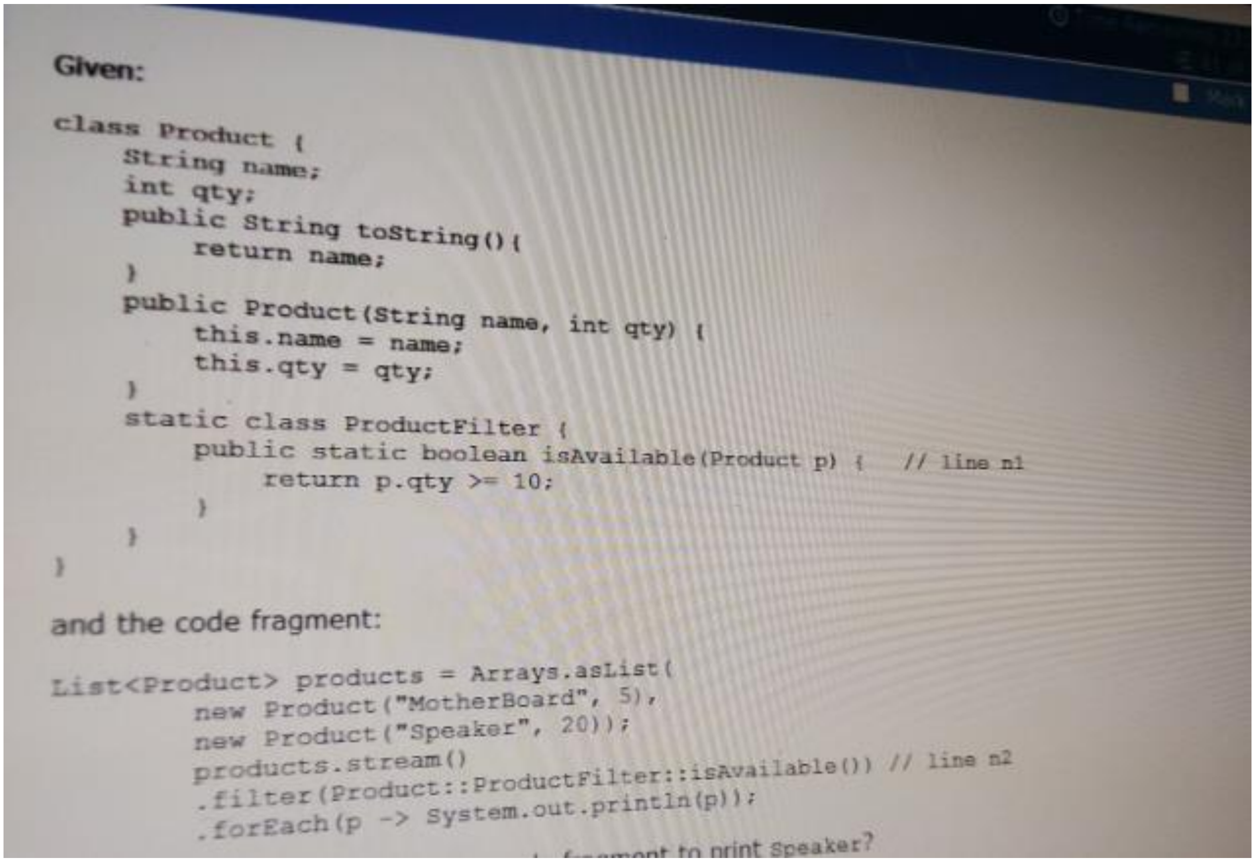
○ B) 4000 : 2000

○ C) 4000 : 1000

○ D) 1000 : 4000

Correct Answer: A

Problem 41:



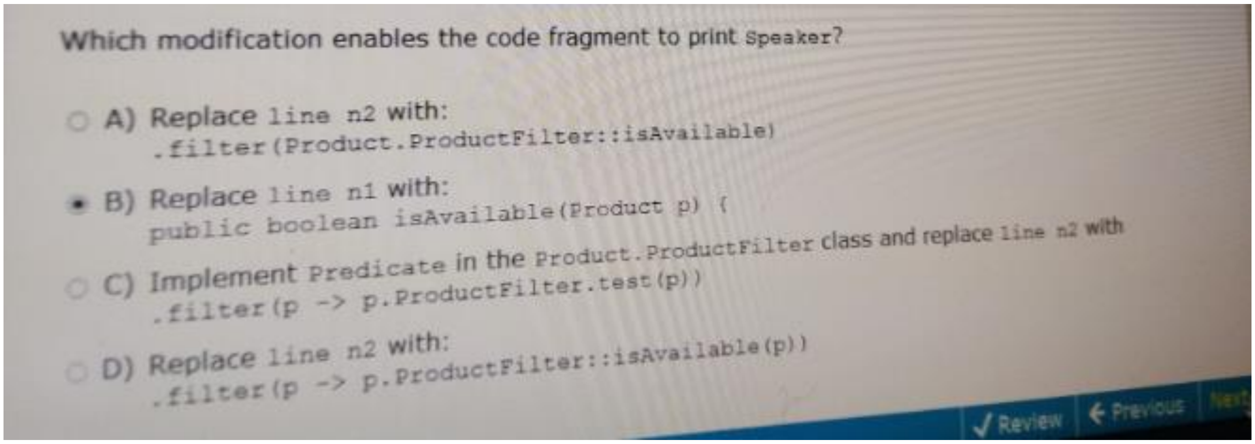
Given:

```
class Product {
    String name;
    int qty;
    public String toString(){
        return name;
    }
    public Product(String name, int qty) {
        this.name = name;
        this.qty = qty;
    }
    static class ProductFilter {
        public static boolean isAvailable(Product p) {   // line n1
            return p.qty >= 10;
        }
    }
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(
        new Product("MotherBoard", 5),
        new Product("Speaker", 20));
        products.stream()
        .filter(Product::ProductFilter::isAvailable()) // line n2
        .forEach(p -> System.out.println(p));
```

fragment to print Speaker?



Which modification enables the code fragment to print Speaker?

○ A) Replace line n2 with:
`.filter(Product.ProductFilter::isAvailable)`

● B) Replace line n1 with:
`public boolean isAvailable(Product p) {`

○ C) Implement Predicate in the Product.ProductFilter class and replace line n2 with
`.filter(p -> p.ProductFilter.test(p))`

○ D) Replace line n2 with:
`.filter(p -> p.ProductFilter::isAvailable(p))`

Correct Answer: B

public class Product

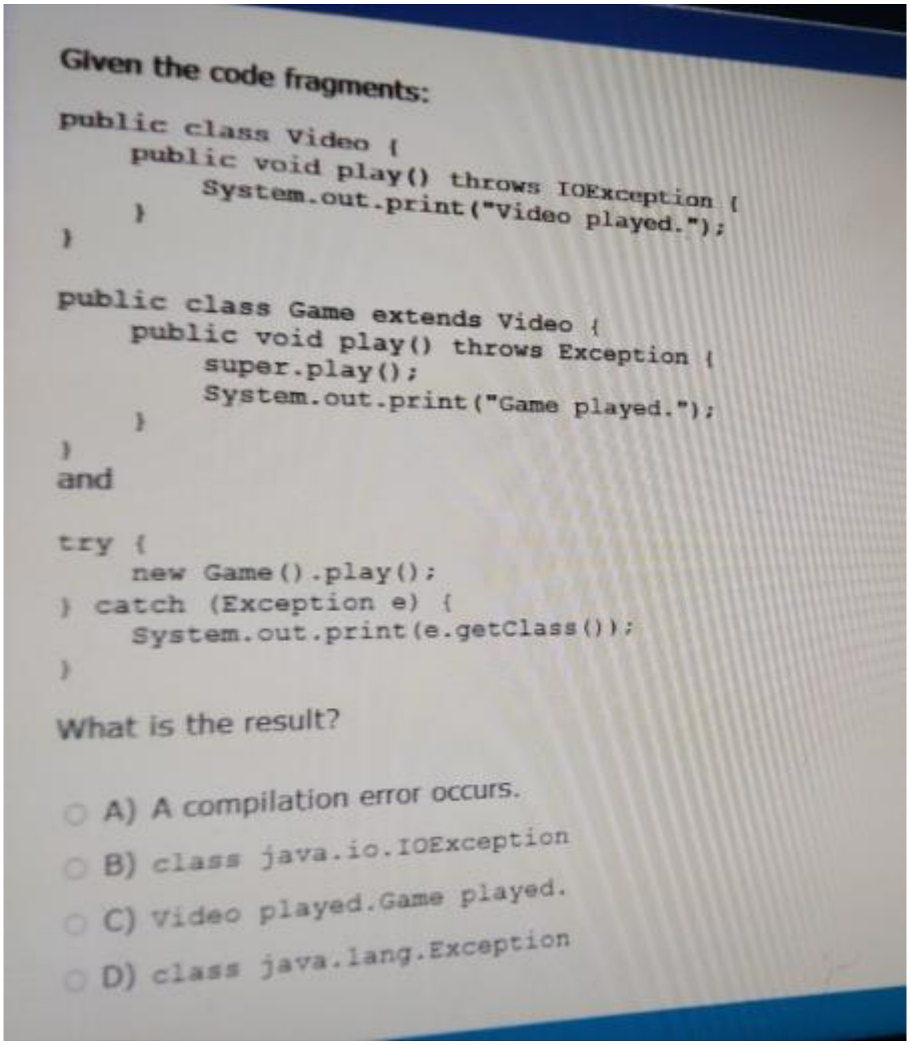
{

String name;

int qty;

```
        public String toString(){
            return name;
        }

        public Product(String name, int qty){
            this.name = name;
            this.qty = qty;
        }

        static class ProductFilter{
            public static boolean isAvailable(Product p){
                return p.qty >= 10;
            }
        }

        List<Product> product = Arrays.asList{
            new Product("MotherBoard",5),
            new Product("Speaker",20);
            product.stream()
            .filter(Product::ProductFilter::isAvailable())
            .forEach(p -> System.out.println(p));
        }
}
```

Problem 42:



Given the code fragments:

```
public class Video {
    public void play() throws IOException {
        System.out.print("Video played.");
    }
}

public class Game extends Video {
    public void play() throws Exception {
        super.play();
        System.out.print("Game played.");
    }
}
```

and

```
try {
    new Game().play();
} catch (Exception e) {
    System.out.print(e.getClass());
}
```

What is the result?

○ A) A compilation error occurs.

○ B) class java.io.IOException

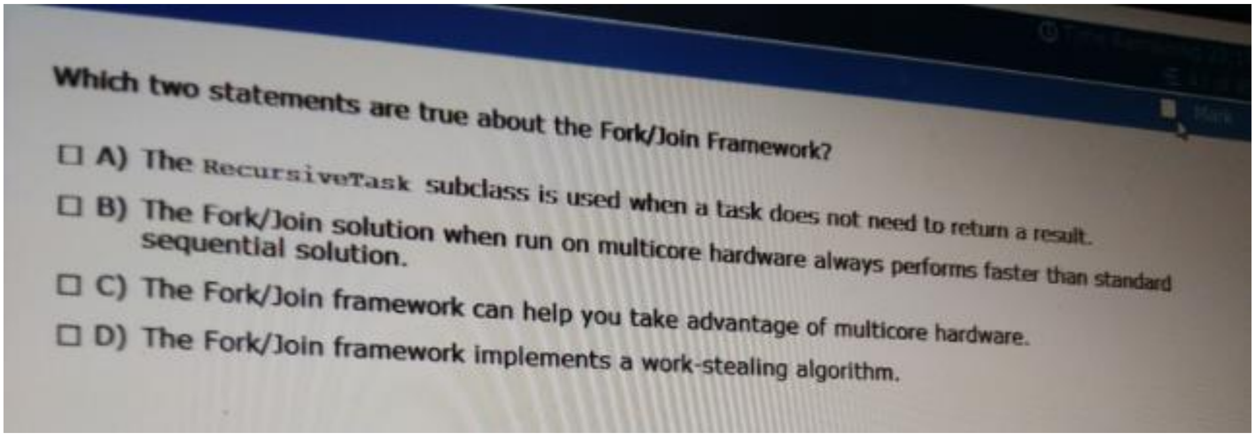
○ C) Video played.Game played.

○ D) class java.lang.Exception

Correct Answer: A

```
package pkg809;

/**
 * Qestion 42
 * @Answer: ?
 */
public class Video {
    public void play() throws IOException{
        System.out.print("Video played.");
    }
}

public class Game extends Video{
   public void play() throws Exception{
      super.play();
      System.out.print("Game played.");
   }
}

try {
   new Game().play();
} catch (Exception e){
   System.out.prin(e.getClass());
}
```

Problem 43:



Which two statements are true about the Fork/Join Framework?

□ A) The RecursiveTask subclass is used when a task does not need to return a result.

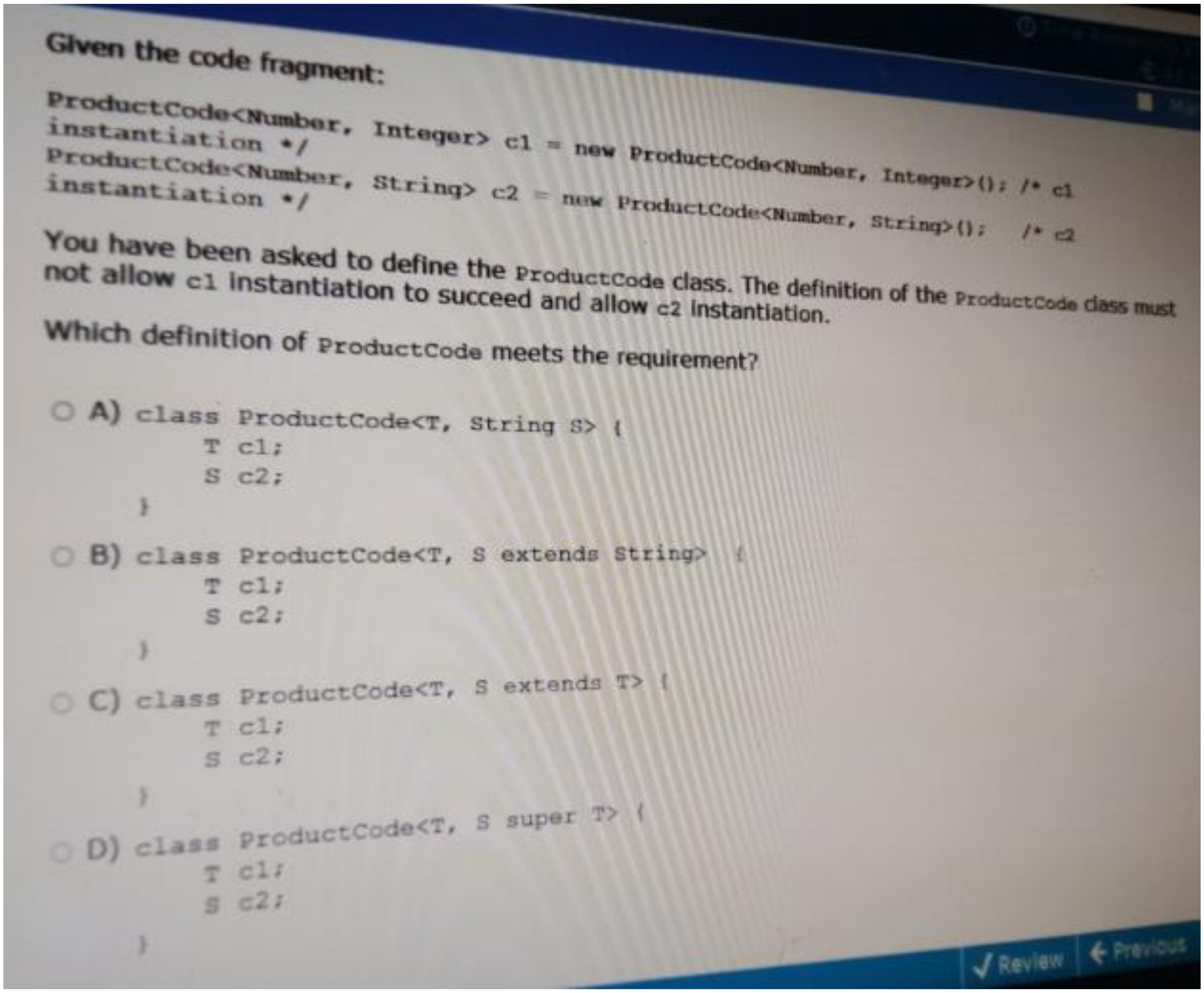
□ B) The Fork/Join solution when run on multicore hardware always performs faster than standard sequential solution.

□ C) The Fork/Join framework can help you take advantage of multicore hardware.

□ D) The Fork/Join framework implements a work-stealing algorithm.

Correct Answer: A, D

Problem 44:



Given the code fragment:

```
ProductCode<Number, Integer> c1 = new ProductCode<Number, Integer>(); /* c1
instantiation */
ProductCode<Number, String> c2 = new ProductCode<Number, String>();   /* c2
instantiation */
```

You have been asked to define the ProductCode class. The definition of the ProductCode class must not allow c1 instantiation to succeed and allow c2 instantiation.

Which definition of ProductCode meets the requirement?

○ A)
```
class ProductCode<T, String S> {
        T c1;
        S c2;
    }
```

○ B)
```
class ProductCode<T, S extends String>  {
        T c1;
        S c2;
    }
```

○ C)
```
class ProductCode<T, S extends T> {
        T c1;
        S c2;
    }
```

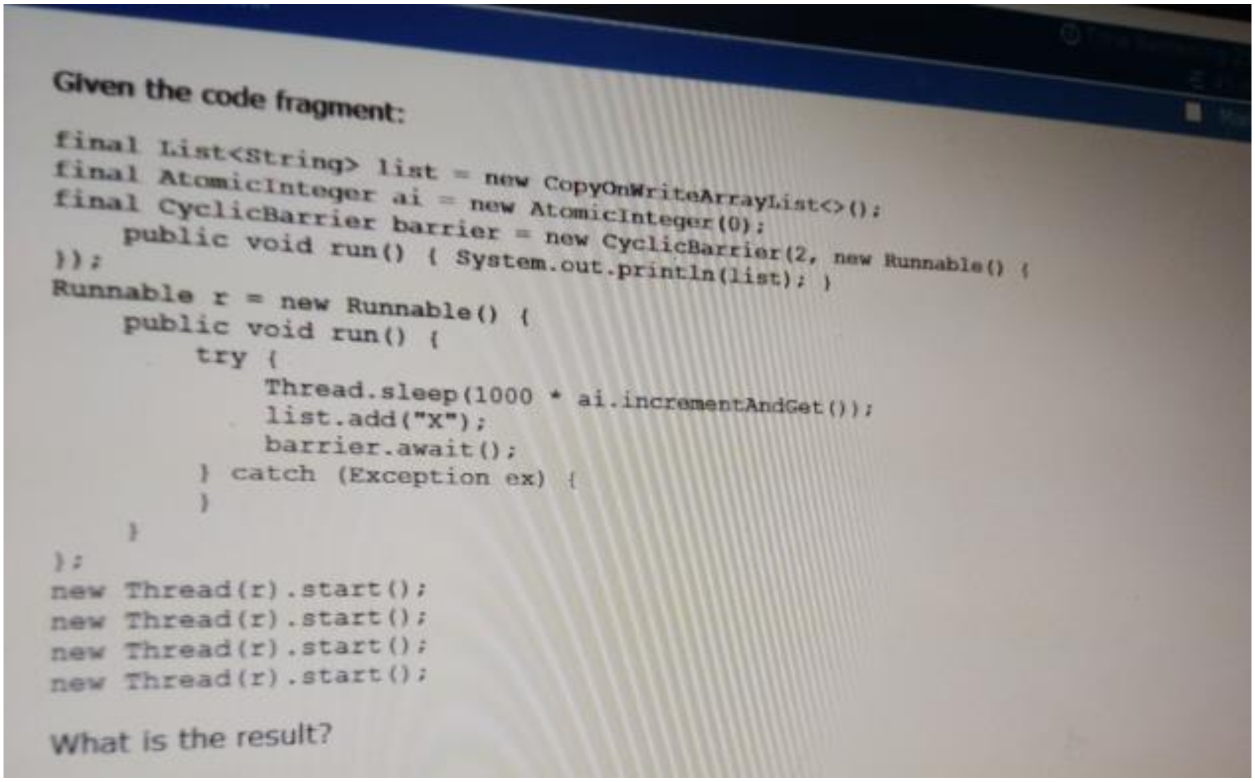
○ D)
```
class ProductCode<T, S super T> {
        T c1;
        S c2;
    }
```

Correct Answer: C

```
package pkg809;

/**
 * Question 44
 * @Answer: ?
 */
public class ProductCode(T, String S) {
   T c1;
   S c2;
```

}

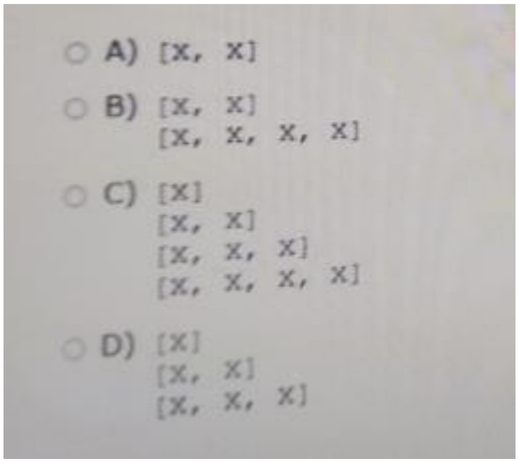
Problem 45:



Given the code fragment:

```
final List<String> list = new CopyOnWriteArrayList<>();
final AtomicInteger ai = new AtomicInteger(0);
final CyclicBarrier barrier = new CyclicBarrier(2, new Runnable() {
     public void run() { System.out.println(list); }
});
Runnable r = new Runnable() {
     public void run() {
          try {
               Thread.sleep(1000 * ai.incrementAndGet());
               list.add("X");
               barrier.await();
          } catch (Exception ex) {
          }
     }
};
new Thread(r).start();
new Thread(r).start();
new Thread(r).start();
new Thread(r).start();
```

What is the result?



○ A) [X, X]

○ B) [X, X]
[X, X, X, X]

○ C) [X]
[X, X]
[X, X, X]
[X, X, X, X]

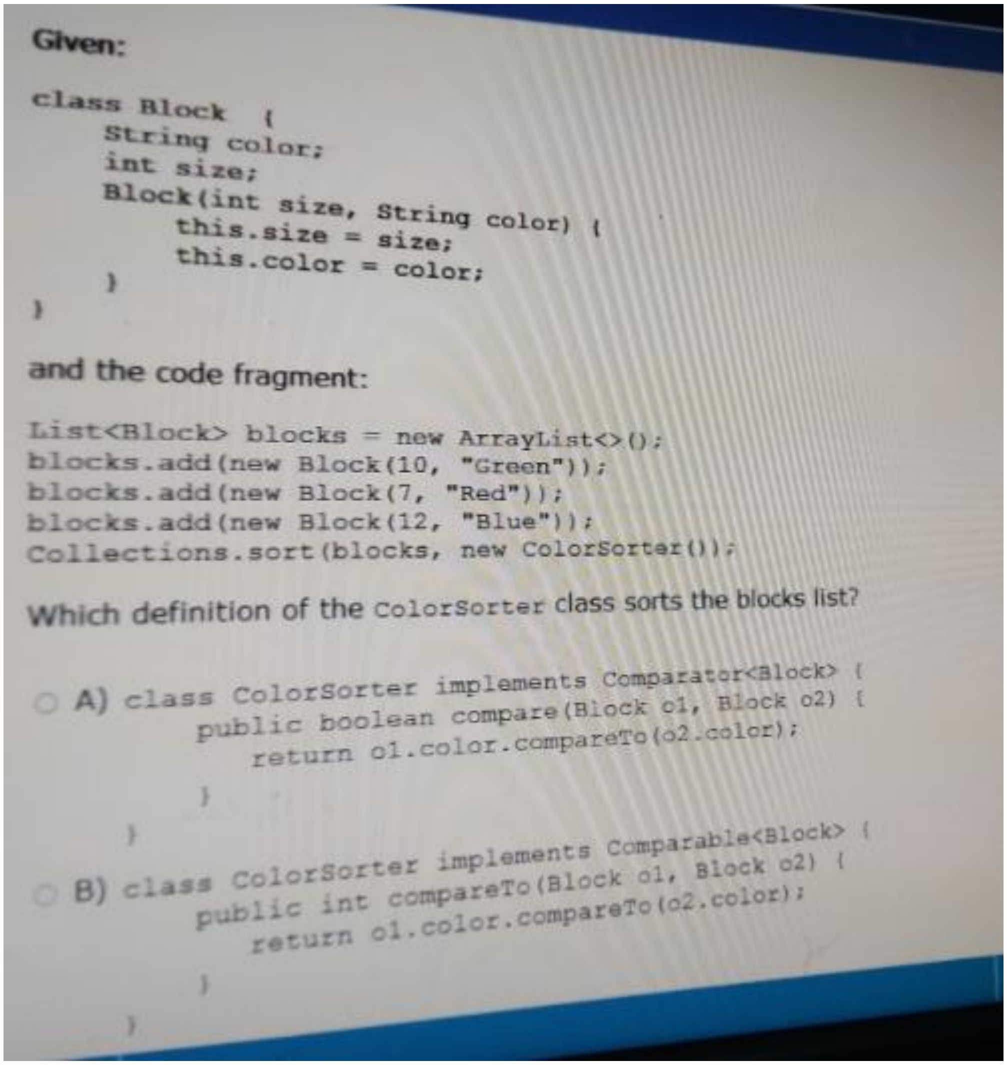
○ D) [X]
[X, X]
[X, X, X]

Correct Answer: B

package pkg809;

```
/**
 * Question 45
 * @Answer: ?
 */
public class Tread {
   final List<String> list = new CopyOnWriteArrayList<>();
   final AtomicInteger ai = new AtomicInteger(0);
   final CyclicBarrier barrier = new CyclicBarrier(2,new Runnable()){
       public void run() {System.out.println(list);}
   });

Runnable r = new Runnable(){
   public void runI(){
       try{
          Thread.sleep(1000 * ai.incrementAndGet());
          list.add("x");
          barrier.await();
       } catch (Exeption ex){


       }
   }
};

new Thread(r).Start();
new Thread(r).Start();
new Thread(r).Start();
new Thread(r).Start();
new Thread(r).Start();
```

Problem 46:



Given:

```
class Block  {
     String color;
     int size;
     Block(int size, String color) {
          this.size = size;
          this.color = color;
     }
}
```

and the code fragment:

```
List<Block> blocks = new ArrayList<>();
blocks.add(new Block(10, "Green"));
blocks.add(new Block(7, "Red"));
blocks.add(new Block(12, "Blue"));
Collections.sort(blocks, new ColorSorter());
```

Which definition of the ColorSorter class sorts the blocks list?

○ A)
```
class ColorSorter implements Comparator<Block> {
     public boolean compare(Block o1, Block o2) {
          return o1.color.compareTo(o2.color);
     }
}
```

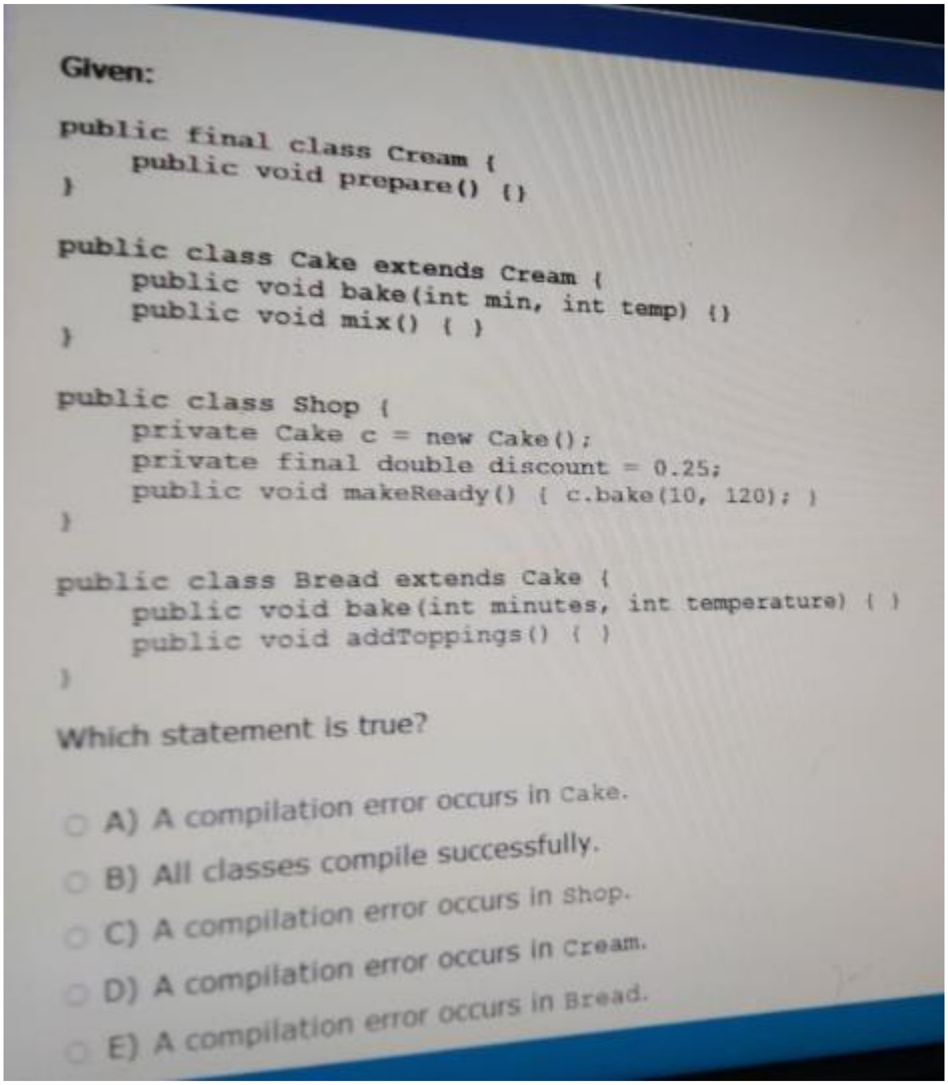
○ B)
```
class ColorSorter implements Comparable<Block> {
     public int compareTo(Block o1, Block o2) {
          return o1.color.compareTo(o2.color);
     }
}
```

Correct Answer: B

package pkg809;

```
/**
 * Question 46
 * @Answer: ?
 */
public class Block {
    String color;
    int size;
    Block(int size, String color){
        this.size = size;
        this.color = color;
    }
}

List<Block> blocks = new ArrayList<>();
blocks.add(new Block(10,"Green"));
blocks.add(new Block(7,"Red"));
blocks.add(new Block(12,"Blue"));
Collections.sort(blocks, new ColorSorterI());
```

Problem 47:



Given:

```
public final class Cream {
     public void prepare() {}
}

public class Cake extends Cream {
     public void bake(int min, int temp) {}
     public void mix() { }
}

public class Shop {
     private Cake c = new Cake();
     private final double discount = 0.25;
     public void makeReady() { c.bake(10, 120); }
}

public class Bread extends Cake {
     public void bake(int minutes, int temperature) { }
     public void addToppings() { }
}
```

Which statement is true?

○ A) A compilation error occurs in `Cake`.

○ B) All classes compile successfully.

○ C) A compilation error occurs in `Shop`.

○ D) A compilation error occurs in `Cream`.

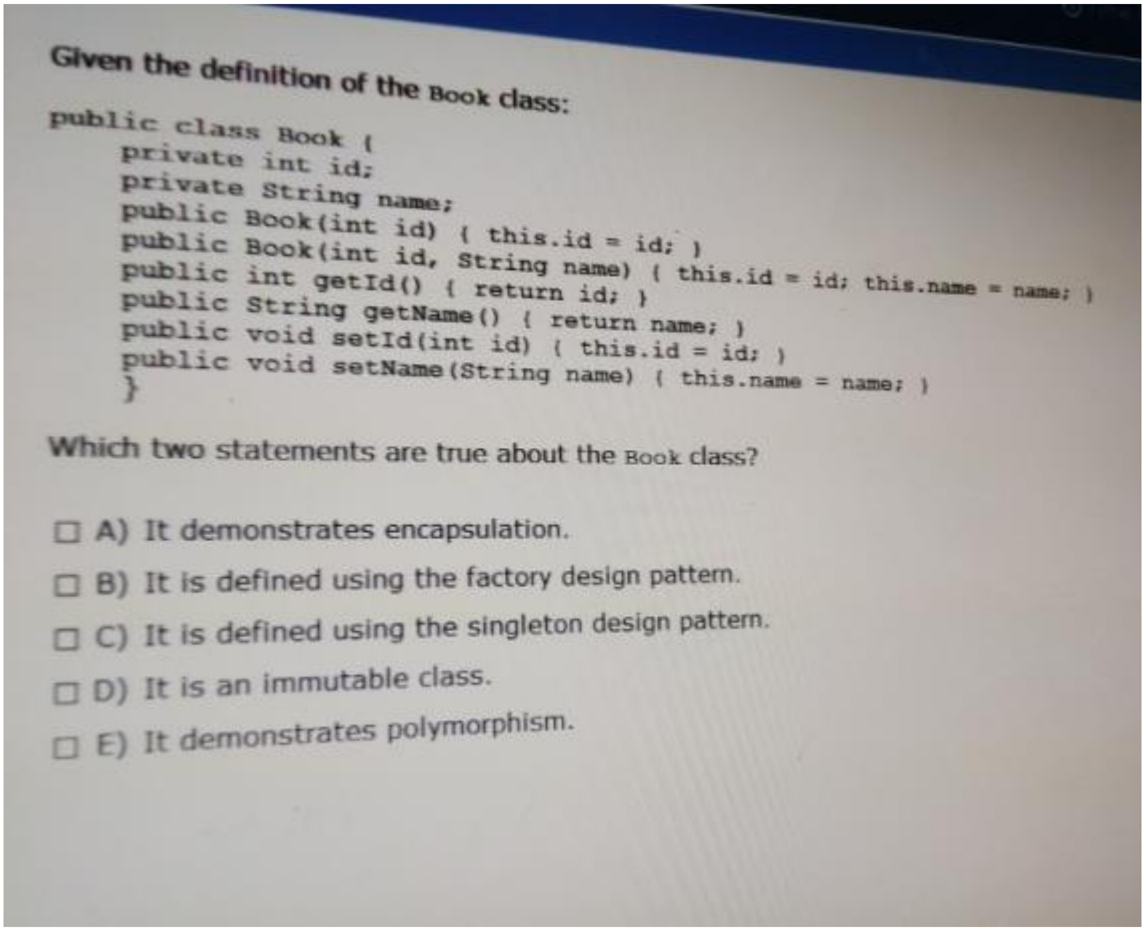
○ E) A compilation error occurs in `Bread`.

Correct Answer: A

package pkg809;

```
/**
 * Question 48
 * @Answer: ?
 */
public class Book {
    private int id;
    private String name;
    public Book(int id) {this.id = id;}
    public Book(int id, String name){this.id; this.name = name}
    public int getId() {return id;}
    public String getName(){return name;}
    public void setIdI(int id){this.id = id;}
    public void setName(String name){this.name = name}
}
```

Problem 48:



Given the definition of the Book class:

```
public class Book {
     private int id;
     private String name;
     public Book(int id) { this.id = id; }
     public Book(int id, String name) { this.id = id; this.name = name; }
     public int getId() { return id; }
     public String getName() { return name; }
     public void setId(int id) { this.id = id; }
     public void setName(String name) { this.name = name; }
     }
```

Which two statements are true about the Book class?

□ A) It demonstrates encapsulation.

□ B) It is defined using the factory design pattern.

□ C) It is defined using the singleton design pattern.

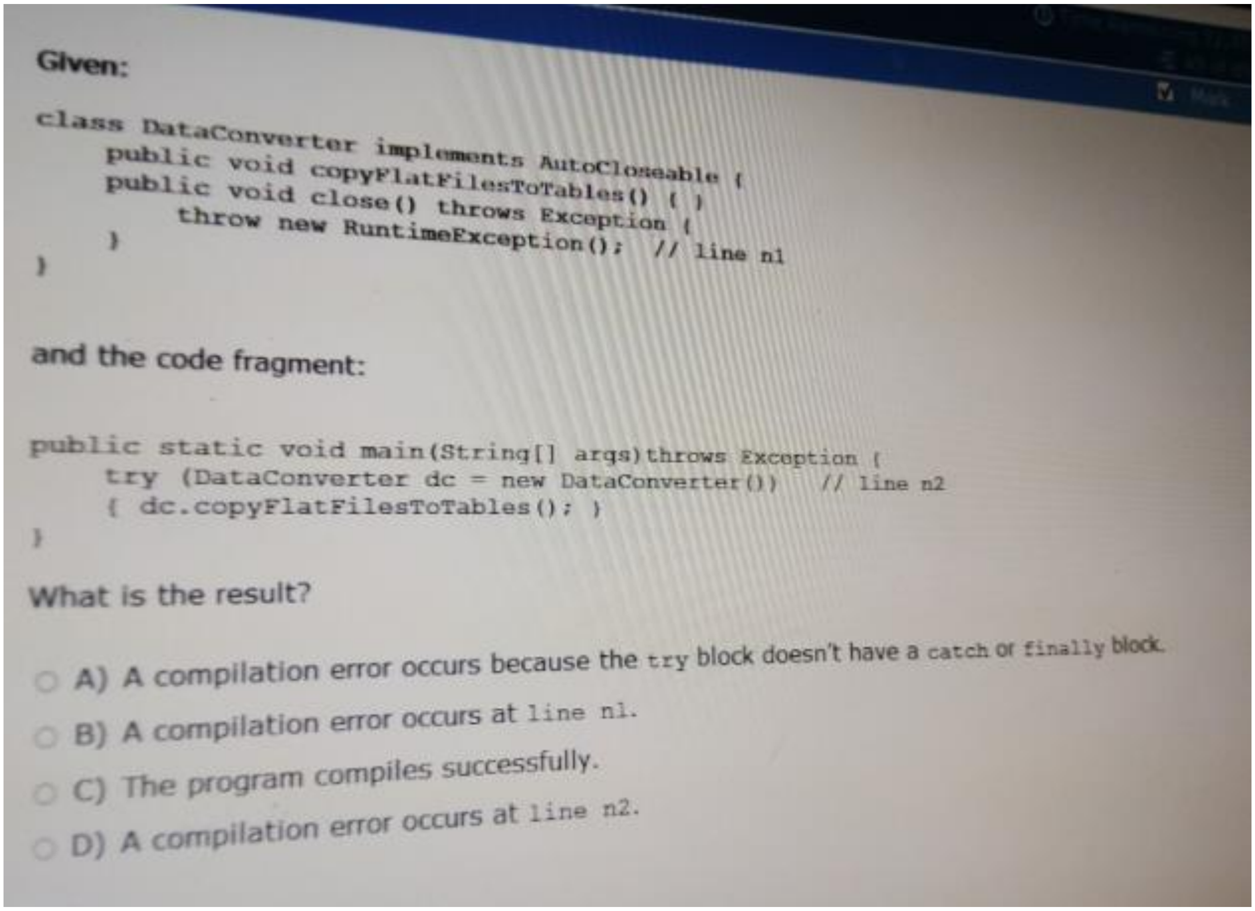
□ D) It is an immutable class.

□ E) It demonstrates polymorphism.

Correct Answer: A

```
package pkg809;

/**
 * Question 48
 * @Answer: ?
 */
public class Book {
    private int id;
    private String name;
    public Book(int id) {this.id = id;}
    public Book(int id, String name){this.id; this.name = name}
    public int getId() {return id;}
    public String getName(){return name;}
    public void setIdI(int id){this.id = id;}
    public void setName(String name){this.name = name}
}
```

Problem 49:



Given:

```
class DataConverter implements AutoCloseable {
    public void copyFlatFilesToTables() { }
    public void close() throws Exception {
        throw new RuntimeException();   // line n1
    }
}
```

and the code fragment:

```
public static void main(String[] args)throws Exception {
    try (DataConverter dc = new DataConverter())    // line n2
    { dc.copyFlatFilesToTables(); }
}
```

What is the result?

○ A) A compilation error occurs because the try block doesn't have a catch or finally block.

○ B) A compilation error occurs at line n1.

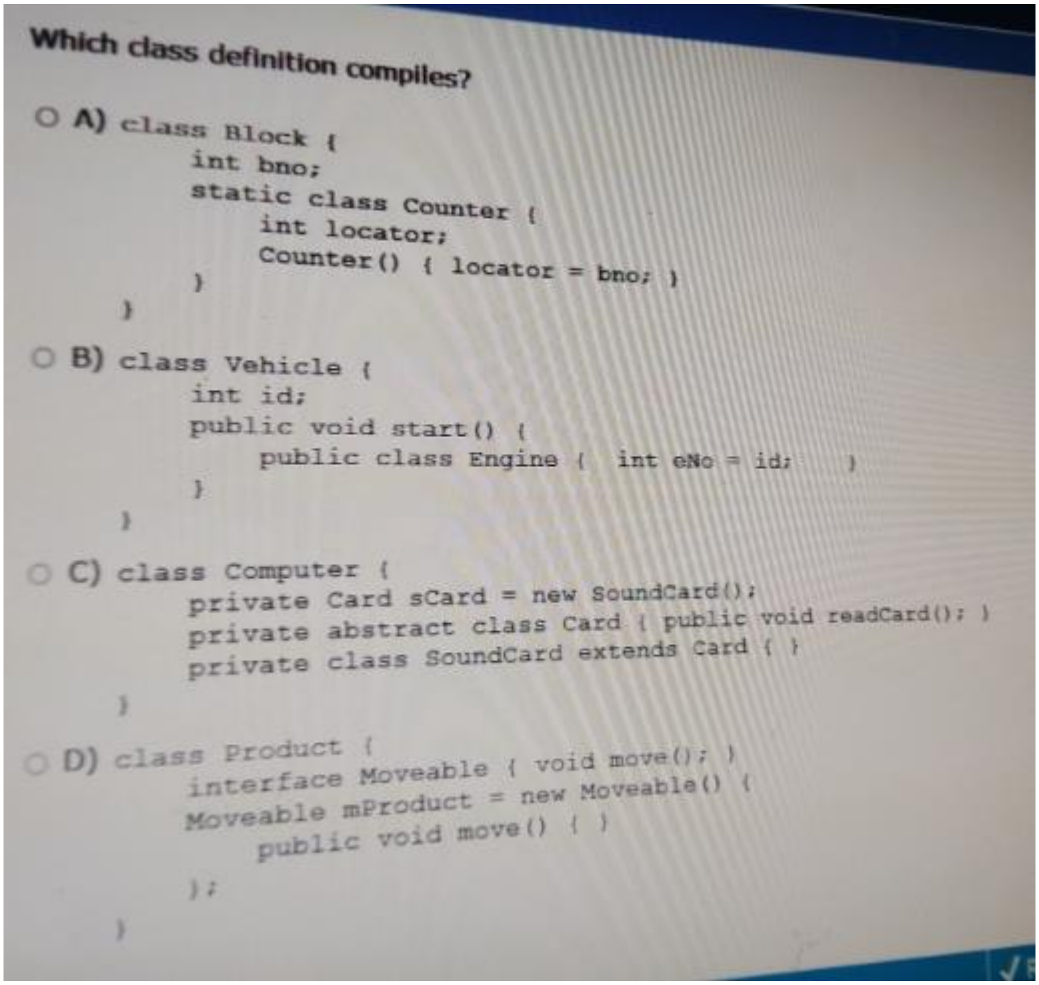
○ C) The program compiles successfully.

○ D) A compilation error occurs at line n2.

Correct Answer: C

```
package pkg809;

/**
 * Qustion 49
 * @Ansewe: ?
 */
class DataConverter implements AutoCloseable {
   public void copyFlatFilesToTables(){}
   public void close() throws Excption {
      throw new RuntimeException();   // line n1
   }
}



public static void main(String[] args) throws Exception{
   try {DataConvater dc = new DataConverter()) // line n2
{dc.copyFlatFilesToTables();}
}
}
```

Problem 50:



Which class definition compiles?

○ A)
```
class Block {
        int bno;
        static class Counter {
            int locator;
            Counter() { locator = bno; }
        }
    }
```

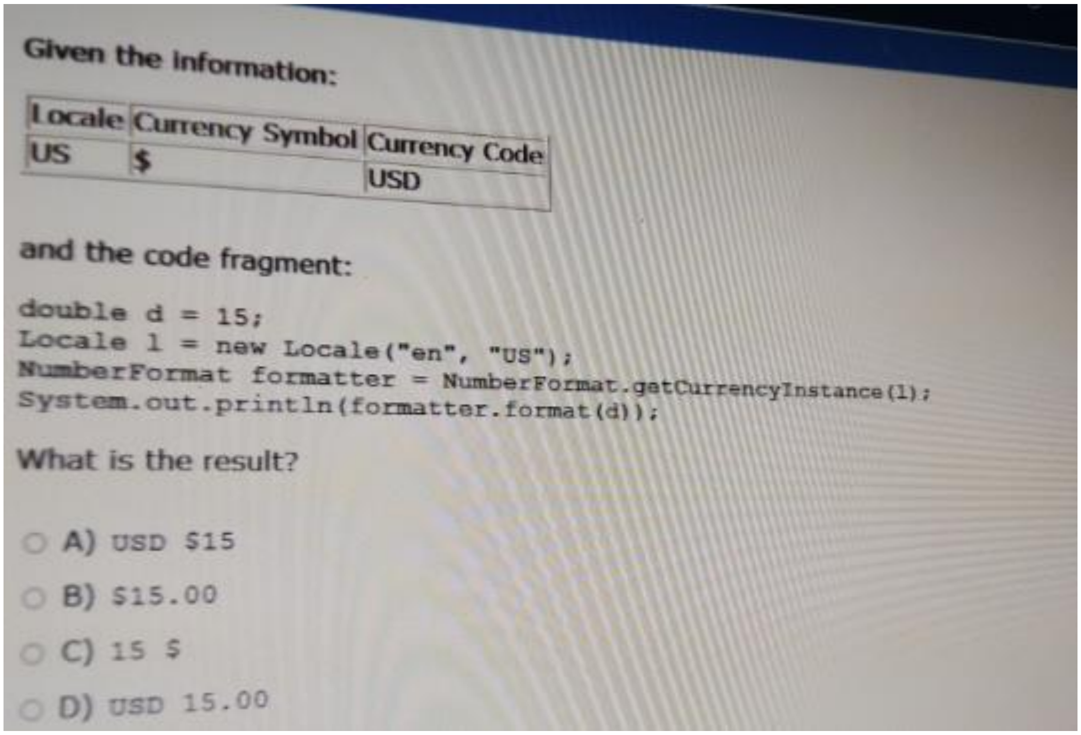
○ B)
```
class Vehicle {
        int id;
        public void start() {
            public class Engine {  int eNo = id;    }
        }
    }
```

○ C)
```
class Computer {
        private Card sCard = new SoundCard();
        private abstract class Card { public void readCard(); }
        private class SoundCard extends Card { }
    }
```

○ D)
```
class Product {
        interface Moveable { void move(); }
        Moveable mProduct = new Moveable() {
            public void move() { }
        };
    }
```

Correct Answer: D

Problem 51:



Given the information:

Locale	Currency Symbol	Currency Code
US	$	USD

and the code fragment:

```
double d = 15;
Locale l = new Locale("en", "US");
NumberFormat formatter = NumberFormat.getCurrencyInstance(l);
System.out.println(formatter.format(d));
```

What is the result?

○ A) USD $15

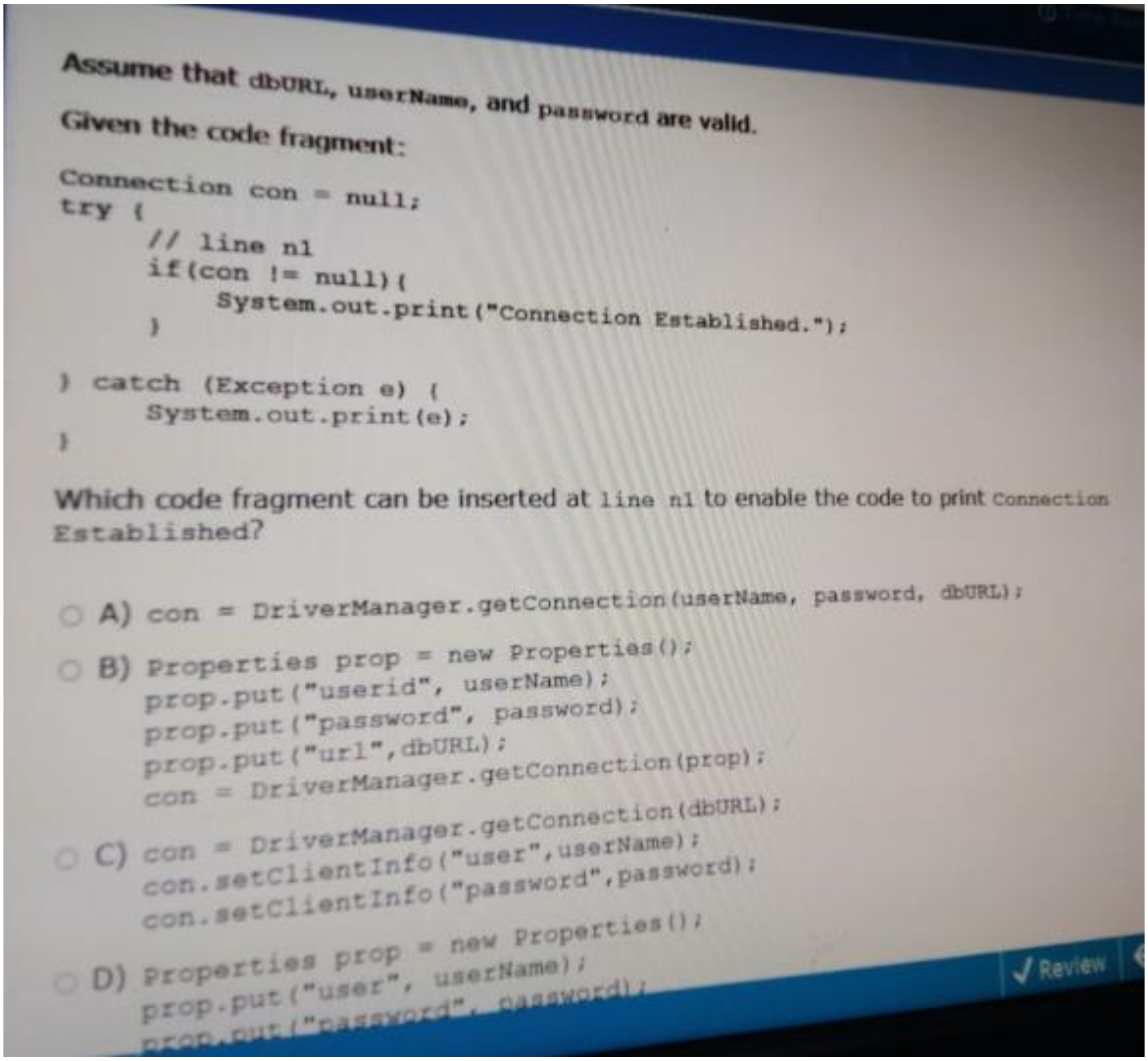
○ B) $15.00

○ C) 15 $

○ D) USD 15.00

Correct Answer: B

Problem 52:



Assume that dbURL, userName, and password are valid.

Given the code fragment:

```
Connection con = null;
try {
        // line n1
        if(con != null){
            System.out.print("Connection Established.");
        }

} catch (Exception e) {
        System.out.print(e);
}
```

Which code fragment can be inserted at line n1 to enable the code to print Connection Established?

○ A) `con = DriverManager.getConnection(userName, password, dbURL);`

○ B)
```
Properties prop = new Properties();
prop.put("userid", userName);
prop.put("password", password);
prop.put("url",dbURL);
con = DriverManager.getConnection(prop);
```

○ C)
```
con = DriverManager.getConnection(dbURL);
con.setClientInfo("user",userName);
con.setClientInfo("password",password);
```

○ D)
```
Properties prop = new Properties();
prop.put("user", userName);
prop.put("password", password);
prop.put(
```

Correct Answer: D

D. Properties prop = new Properties();

prop.put("user", userName);

prop.put("password", password);

con = DriverManager.getConnection(dbURL, prop);

package com.coderbd.Q52;

import java.sql.Connection;

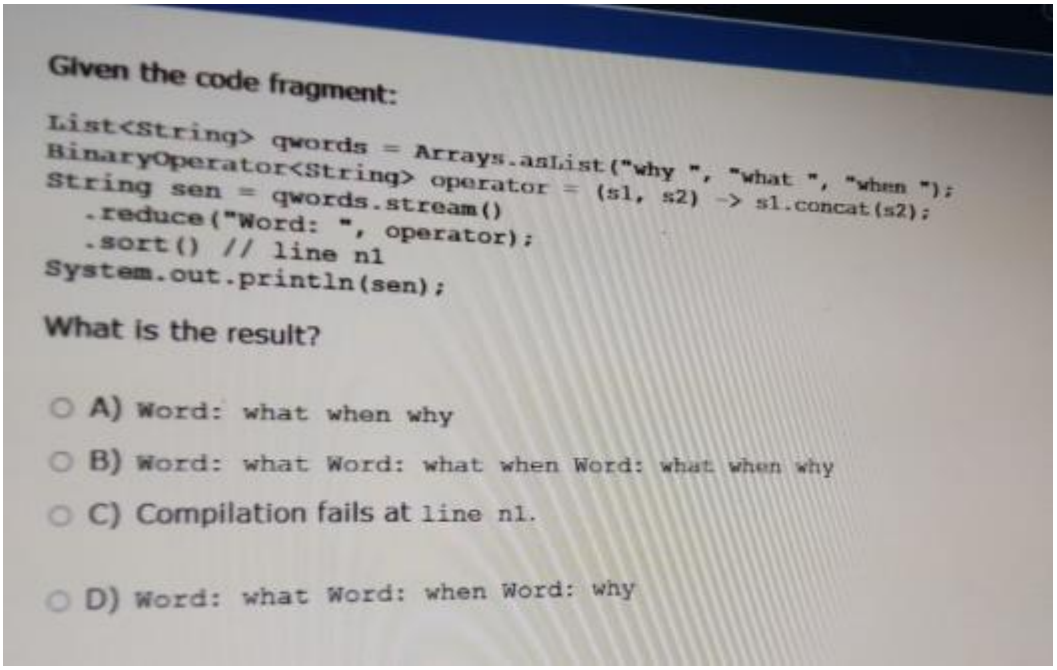
import java.sql.DriverManager;

import java.util.Properties;

```
public class Test {
   public static void main(String[] args) {
      Connection con=null;
      try{
         // line n1
    //A//     con=DriverManager.getConnection(username,password,dbURL);

//  B///         Properties prop=new Properties();
//               prop.put("userid", userName);
//               prop.put("Password", password);
//               prop.put("url", dbURL);
//               con=DriverManager.getConnection(prop);



//   C//             con=DriverManager.getConnection(dbURL);
//                con.setClientInfo("user",userName);
//                con.setClientInfo("password",password);
//
//   D//               Properties prop=new Properties();
//                  prop.put("user", userName);
//                  prop.put("password", password);
         if(con!=null){
            System.out.println("Connection Established.");
         }
      }catch(Exception e){
         System.out.println(e);
      }
   }
```

}

Problem 53:



Given the code fragment:

```
List<String> qwords = Arrays.asList("why ", "what ", "when ");
BinaryOperator<String> operator = (s1, s2) -> s1.concat(s2);
String sen = qwords.stream()
  .reduce("Word: ", operator);
  .sort() // line n1
System.out.println(sen);
```

What is the result?

○ A) `Word: what when why`

○ B) `Word: what Word: what when Word: what when why`

○ C) Compilation fails at `line n1`.

○ D) `Word: what Word: when Word: why`

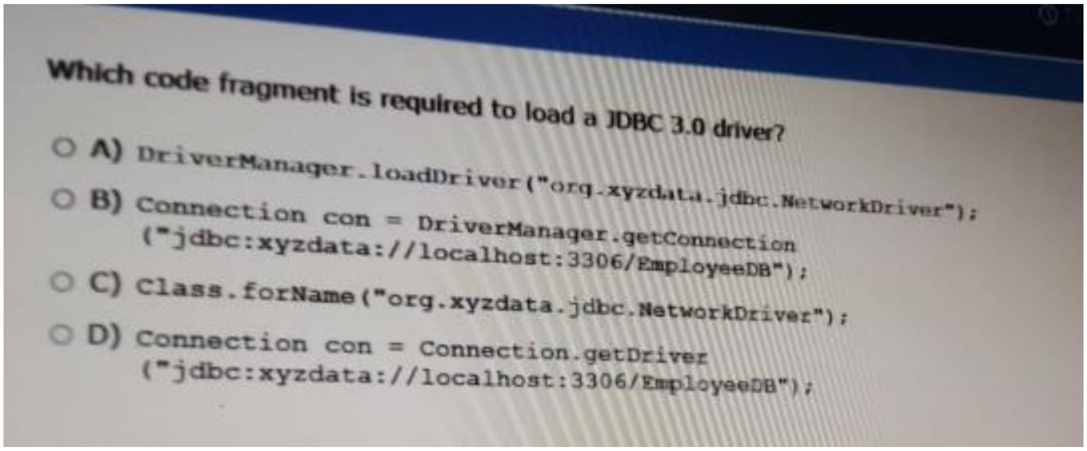
Correct Answer: C

package com.coderbd.Q53;

```
import java.util.Arrays;
import java.util.List;
import java.util.function.BinaryOperator;

public class Test {
    public static void main(String[] args) {
        List<String> qwords=Arrays.asList("why","what","when");
        BinaryOperator<String> operator=(s1,s2) -> s1.concat(s2);
        String sen=qwords.stream()
```

```
            .reduce("Word", operator);

            .sort()                                ///line n1

        System.out.println(sen);


    }
 ///   Ans: C

}
```

Problem 54:



Which code fragment is required to load a JDBC 3.0 driver?

○ A) `DriverManager.loadDriver("org.xyzdata.jdbc.NetworkDriver");`

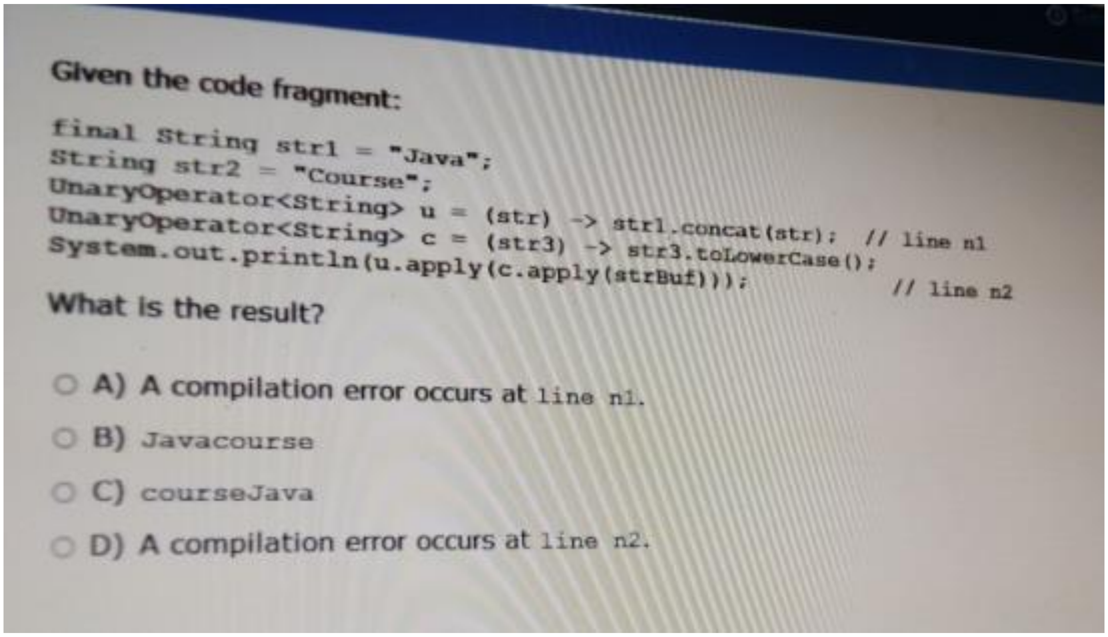
○ B)
```
Connection con = DriverManager.getConnection
 ("jdbc:xyzdata://localhost:3306/EmployeeDB");
```

○ C) `Class.forName("org.xyzdata.jdbc.NetworkDriver");`

○ D)
```
Connection con = Connection.getDriver
 ("jdbc:xyzdata://localhost:3306/EmployeeDB");
```

Correct Answer: C

Problem 55:



Given the code fragment:

```
final String str1 = "Java";
String str2 = "Course";
UnaryOperator<String> u = (str) -> str1.concat(str);  // line n1
UnaryOperator<String> c = (str3) -> str3.toLowerCase();
System.out.println(u.apply(c.apply(strBuf)));           // line n2
```

What is the result?

○ A) A compilation error occurs at line n1.

○ B) Javacourse

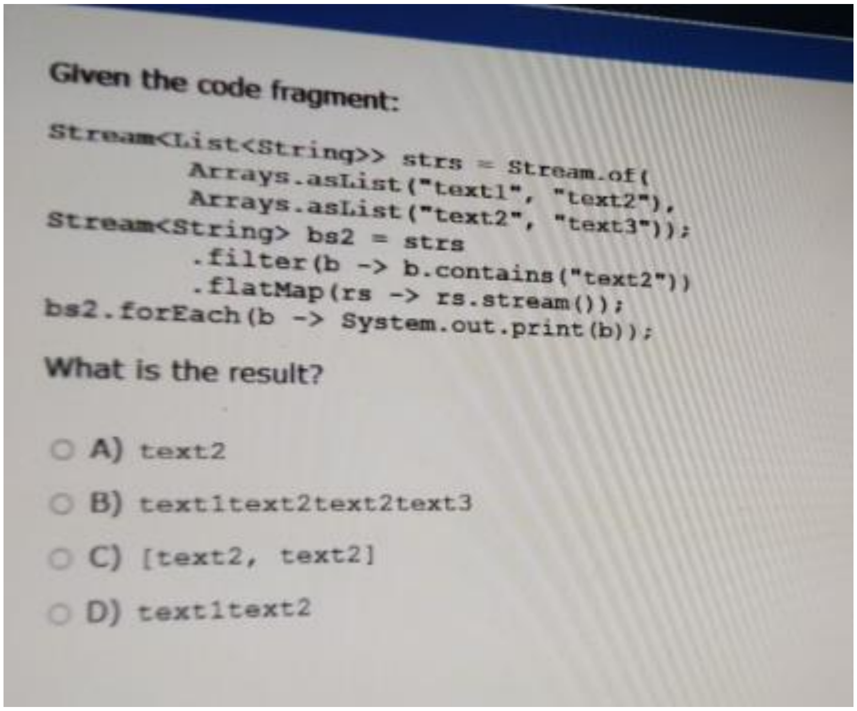
○ C) courseJava

○ D) A compilation error occurs at line n2.

Correct Answer: D

```
package com.coderbd.Q55;


import java.util.function.UnaryOperator;


public class Test {
    public static void main(String[] args) {
        final String str1="Java";
        String str2="Course";
        UnaryOperator<String> u=(str) -> str1.concat(str);      //line n1
        UnaryOperator<String> c=(str3) -> str3.toLowerCase();
        System.out.println(u.apply(c.apply(strBuf)));        // line n2
    }
    ///// Ans: D
}
```

Problem 56:



Given the code fragment:

```
Stream<List<String>> strs = Stream.of(
            Arrays.asList("text1", "text2"),
            Arrays.asList("text2", "text3"));
Stream<String> bs2 = strs
            .filter(b -> b.contains("text2"))
            .flatMap(rs -> rs.stream());
bs2.forEach(b -> System.out.print(b));
```

What is the result?

○ A) text2

○ B) text1text2text2text3

○ C) [text2, text2]

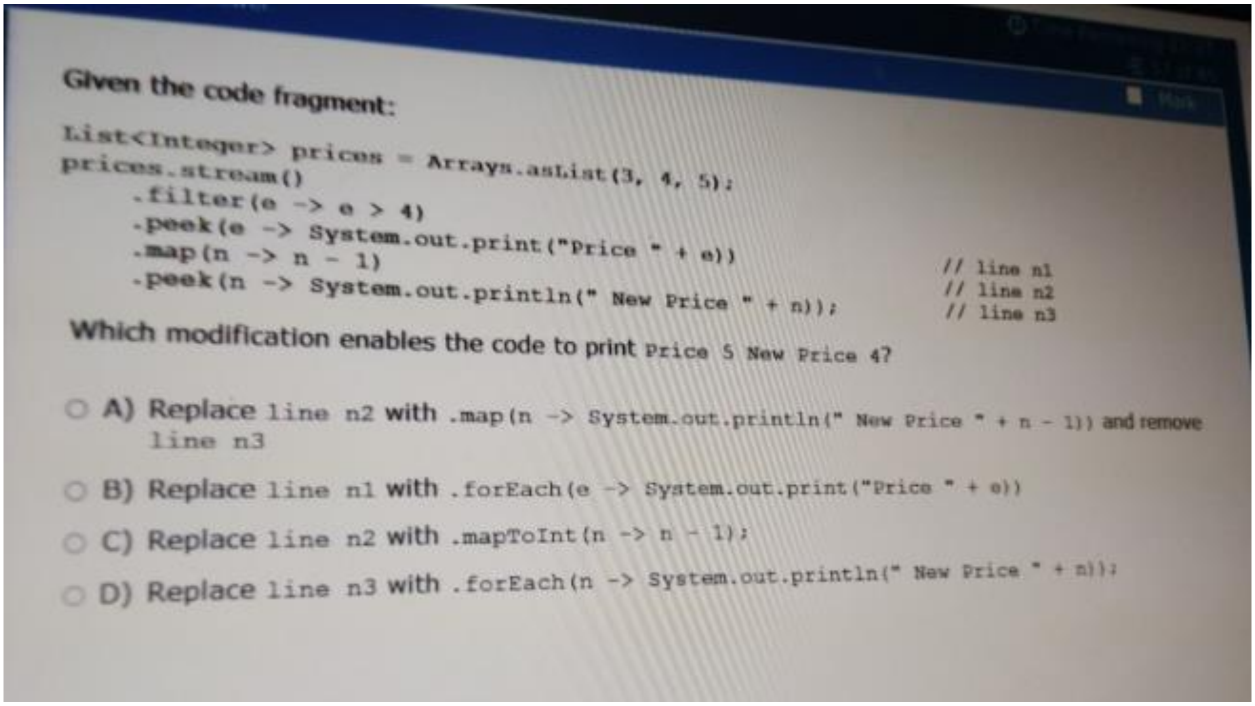
○ D) text1text2

Correct Answer: B

```
package com.coderbd.Q56;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Stream;

/**
 *
 * @author Touhid
 */
public class Test {
   public static void main(String[] args) {
      Stream <List<String>> strs=Stream.of(Arrays.asList("text1","text2"),
            Arrays.asList("text2","Text3"));
```

```
        Stream<String> bs2=strs
                .filter(b -> b.contains("text2"))
                .flatMap(rs -> rs.stream());
        bs2.forEach(b -> System.out.print(b));

    }
    // Ans B
}
```

Problem 57:



Given the code fragment:

```
List<Integer> prices = Arrays.asList(3, 4, 5);
prices.stream()
      .filter(e -> e > 4)
      .peek(e -> System.out.print("Price " + e))          // line n1
      .map(n -> n - 1)                                      // line n2
      .peek(n -> System.out.println(" New Price " + n));   // line n3
```

Which modification enables the code to print `Price 5 New Price 4`?

○ A) Replace `line n2` with `.map(n -> System.out.println(" New Price " + n - 1))` and remove `line n3`

○ B) Replace `line n1` with `.forEach(e -> System.out.print("Price " + e))`

○ C) Replace `line n2` with `.mapToInt(n -> n - 1);`

○ D) Replace `line n3` with `.forEach(n -> System.out.println(" New Price " + n));`

Correct Answer: D

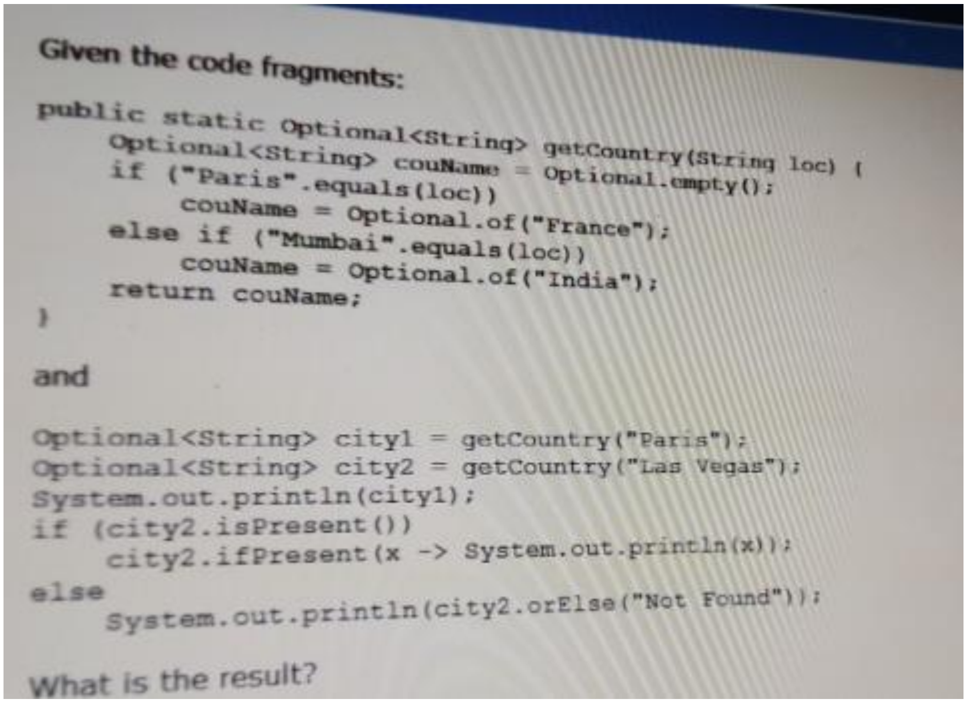
package com.coderbd.Q57;

import java.util.Arrays;

import java.util.List;

```
public class Test {
    public static void main(String[] args) {
        List<Integer> prices=Arrays.asList(3,4,5);
        prices.stream()
                .filter(e -> e > 4)
                .peek(e -> System.out.print("Price "+e))                ////line n1
                .map(n -> n-1)                                  //line n2
                .forEach(n -> System.out.println(" New Price " + n));       //line n3
    }
   //Ans D
}
```

Problem 58:



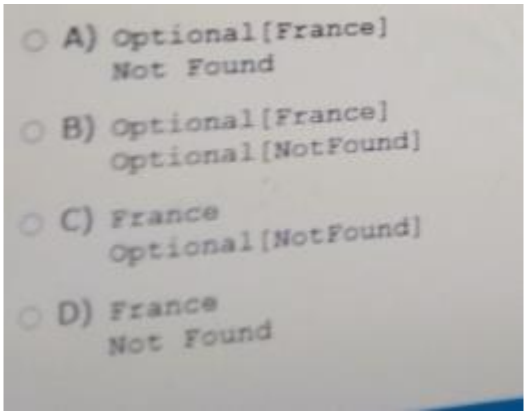
Given the code fragments:

```
public static Optional<String> getCountry(String loc) {
    Optional<String> couName = Optional.empty();
    if ("Paris".equals(loc))
        couName = Optional.of("France");
    else if ("Mumbai".equals(loc))
        couName = Optional.of("India");
    return couName;
}
```

and

```
Optional<String> city1 = getCountry("Paris");
Optional<String> city2 = getCountry("Las Vegas");
System.out.println(city1);
if (city2.isPresent())
    city2.ifPresent(x -> System.out.println(x));
else
    System.out.println(city2.orElse("Not Found"));
```

What is the result?



○ A) Optional[France]
Not Found

○ B) Optional[France]
Optional[NotFound]

○ C) France
Optional[NotFound]

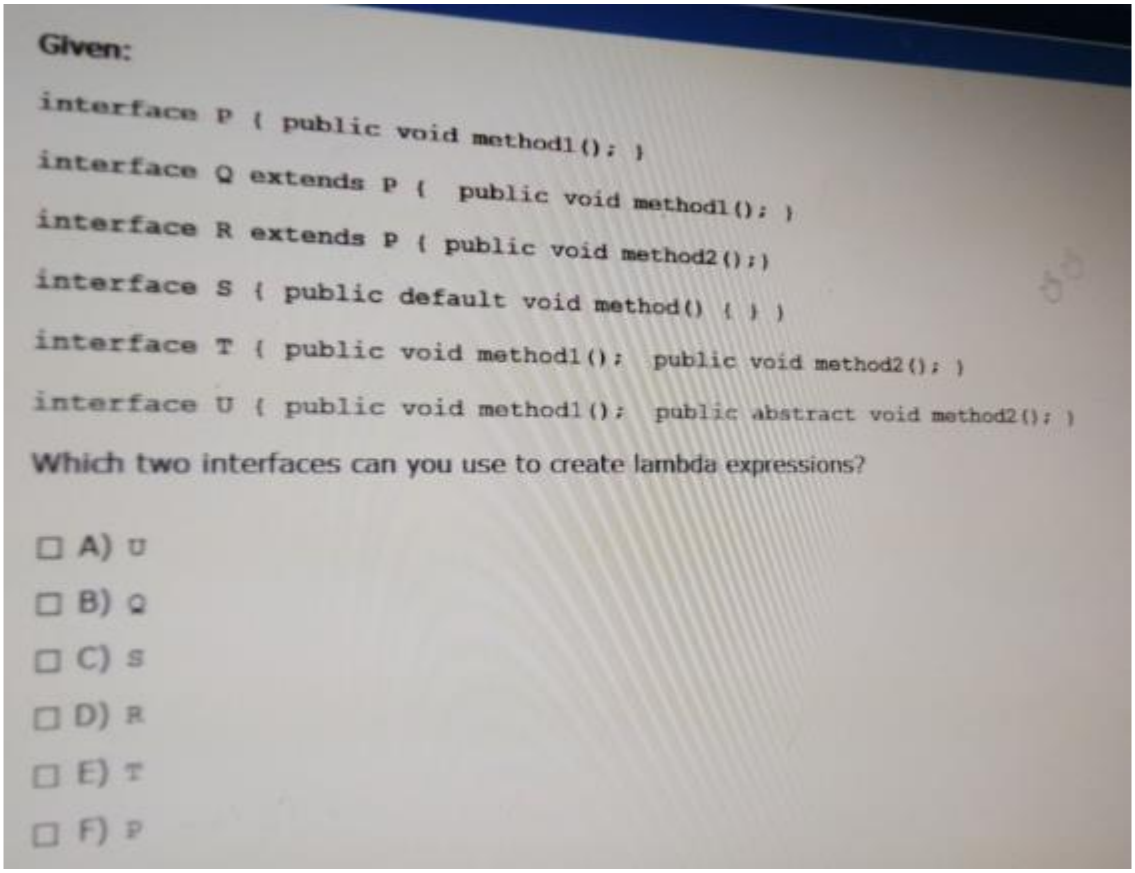
○ D) France
Not Found

Correct Answer: A

```
package com.coderbd.Q58;

import java.util.Optional;

public class Test {
  public static Optional<String> getCountry(String loc){
    Optional<String> couName=Optional.empty();
    if("Paris".equals(loc)){
      couName=Optional.of("France");
    }else if("Mumbai".equals(loc))
    {
      couName=Optional.of("India");
    }

    return couName;
  }
  public static void main(String[] args) {
    Optional<String> city1 = getCountry("Paris");
    Optional<String> city2 = getCountry("Las Vegas");
    System.out.println(city1);
```

```
        if(city1.isPresent()){
          city2.ifPresent(x -> System.out.println(x));
        }else{
          System.out.println(city2.orElse("Not Found"));
        }
    }
  //  Ans: Confuse A
}
```

Problem 59:



Given:

```
interface P { public void method1(); }
interface Q extends P {  public void method1(); }
interface R extends P { public void method2();}
interface S { public default void method() { } }
interface T { public void method1();  public void method2(); }
interface U { public void method1();  public abstract void method2(); }
```

Which two interfaces can you use to create lambda expressions?

□ A) U

□ B) Q

□ C) S

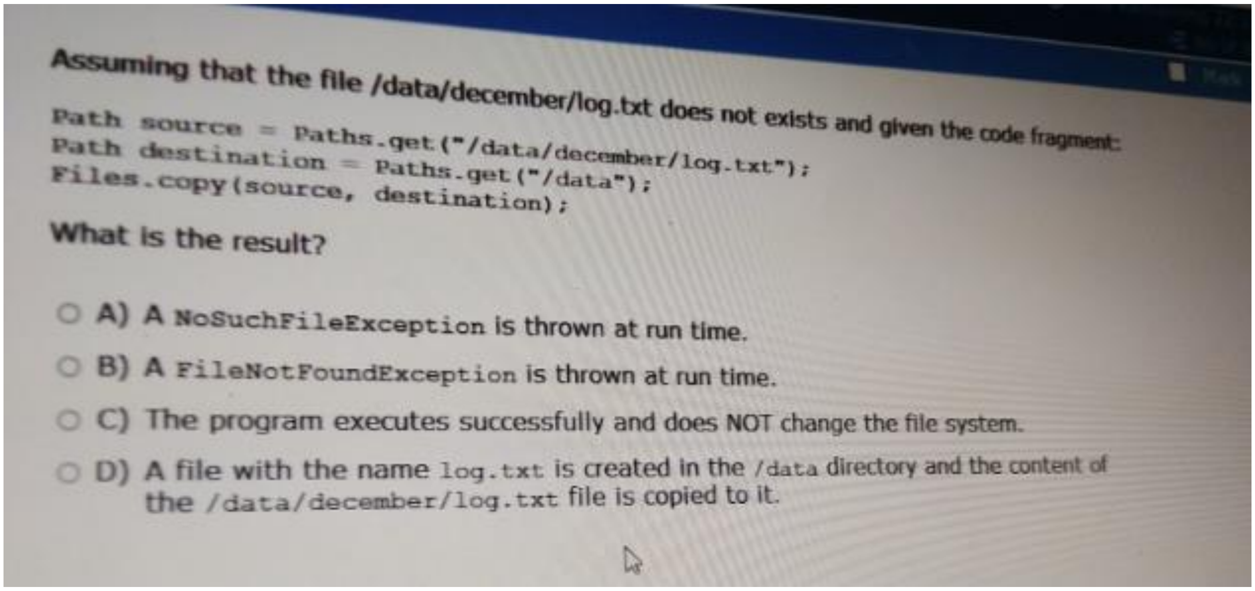
□ D) R

□ E) T

□ F) P

Correct Answer: C, F

Problem 60:



Assuming that the file /data/december/log.txt does not exists and given the code fragment:

```
Path source = Paths.get("/data/december/log.txt");
Path destination = Paths.get("/data");
Files.copy(source, destination);
```

What is the result?

○ A) A `NoSuchFileException` is thrown at run time.

○ B) A `FileNotFoundException` is thrown at run time.

○ C) The program executes successfully and does NOT change the file system.

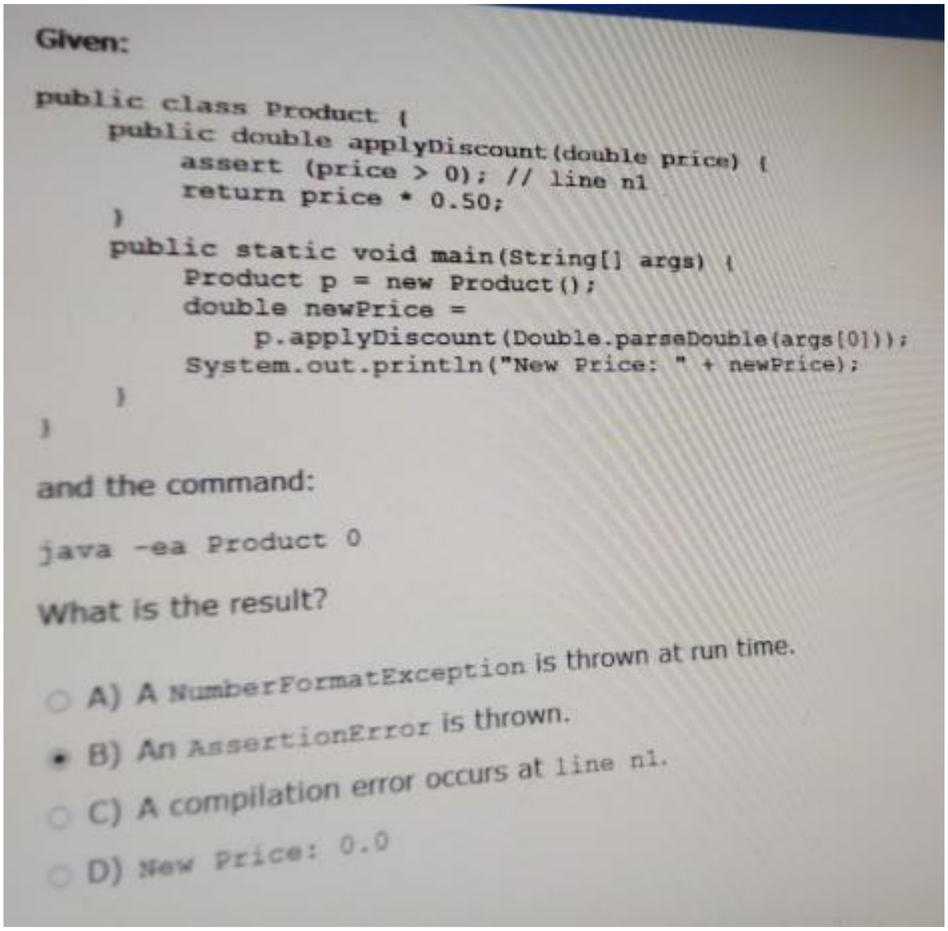
○ D) A file with the name `log.txt` is created in the `/data` directory and the content of the `/data/december/log.txt` file is copied to it.

Correct Answer: A

```
package com.coderbd.Q60;


import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;


/**

 *

 * @author Touhid

 */

public class Test {

    public static void main(String[] args) throws IOException {

        Path source=Paths.get("/data/decembar/log.txt");

        Path destination=Paths.get("/data");

        Files.copy(source, destination);
```

```
   }
   //Ans: A   NoSuchFileException is thrown at runtime


 /////////////  ------>>> NoSuchFileException : \data\decembar\log.txt
}
```

Problem 61:



Given:

```
public class Product {
    public double applyDiscount(double price) {
        assert (price > 0); // line n1
        return price * 0.50;
    }
    public static void main(String[] args) {
        Product p = new Product();
        double newPrice =
            p.applyDiscount(Double.parseDouble(args[0]));
        System.out.println("New Price: " + newPrice);
    }
}
```

and the command:

```
java -ea Product 0
```

What is the result?

○ A) A NumberFormatException is thrown at run time.

● B) An AssertionError is thrown.

○ C) A compilation error occurs at line n1.

○ D) New Price: 0.0

Correct Answer: B

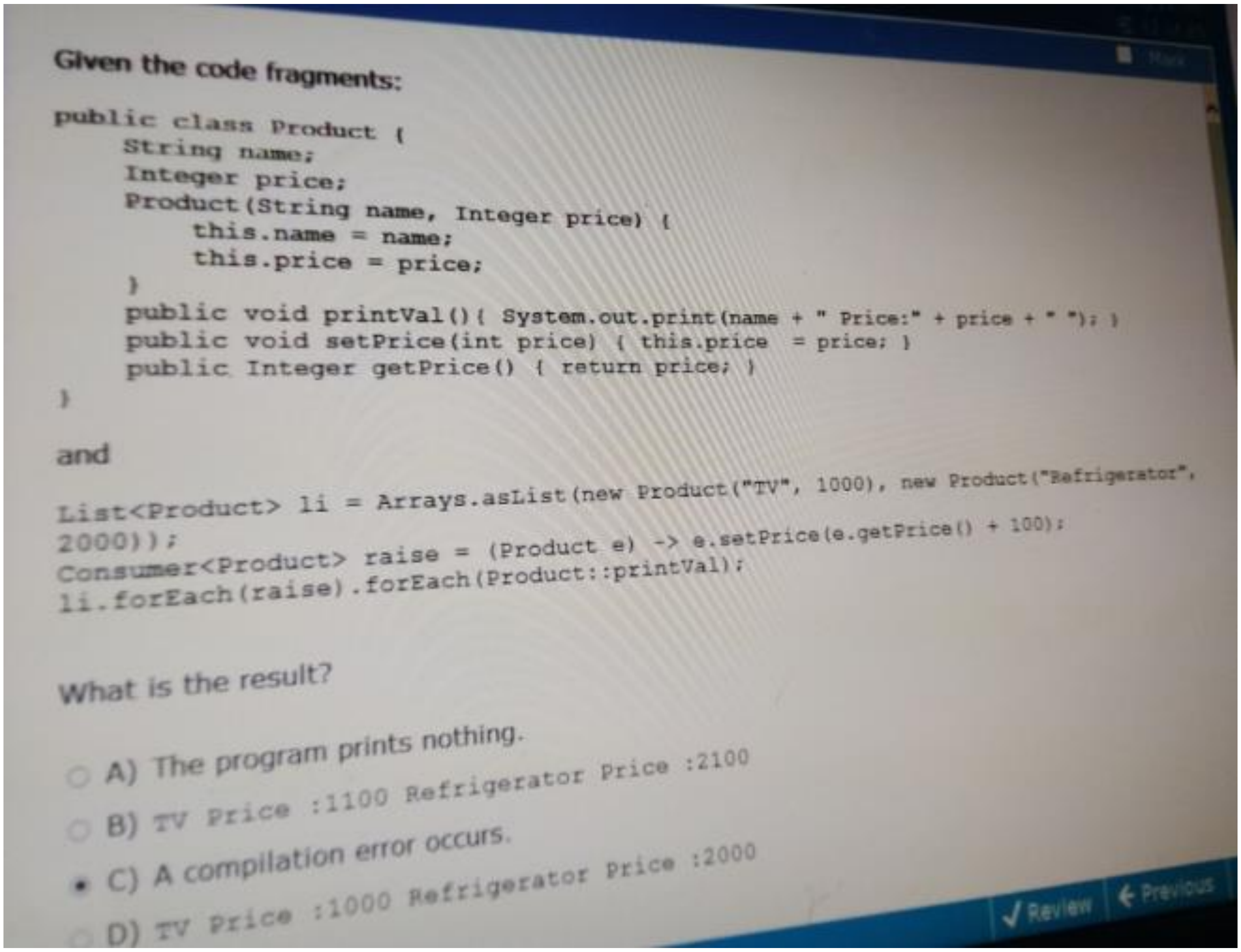
package com.coderbd.q61;

/**

 * Answer: B

```
 */
public class Product {
    public double applyDiscount(double price) {
        assert (price > 0);
        return price * 0.50;
    }

    public static void main(String[] args) {
        Product p = new Product();
        double newPrice = p.applyDiscount(Double.parseDouble(args[0]));
        System.out.println("New price: " + newPrice);
    }
}
```

Problem 62:



Given the code fragments:

```
public class Product {
     String name;
     Integer price;
     Product(String name, Integer price) {
          this.name = name;
          this.price = price;
     }
     public void printVal(){ System.out.print(name + " Price:" + price + " "); }
     public void setPrice(int price) { this.price  = price; }
     public Integer getPrice() { return price; }
}
```

and

```
List<Product> li = Arrays.asList(new Product("TV", 1000), new Product("Refrigerator",
2000));
Consumer<Product> raise = (Product e) -> e.setPrice(e.getPrice() + 100);
li.forEach(raise).forEach(Product::printVal);
```

What is the result?

○ A) The program prints nothing.

○ B) TV Price :1100 Refrigerator Price :2100

● C) A compilation error occurs.

○ D) TV Price :1000 Refrigerator Price :2000

Correct Answer: C

package com.coderbd.q62;

import java.util.Arrays;

import java.util.List;

import java.util.function.Consumer;

/**

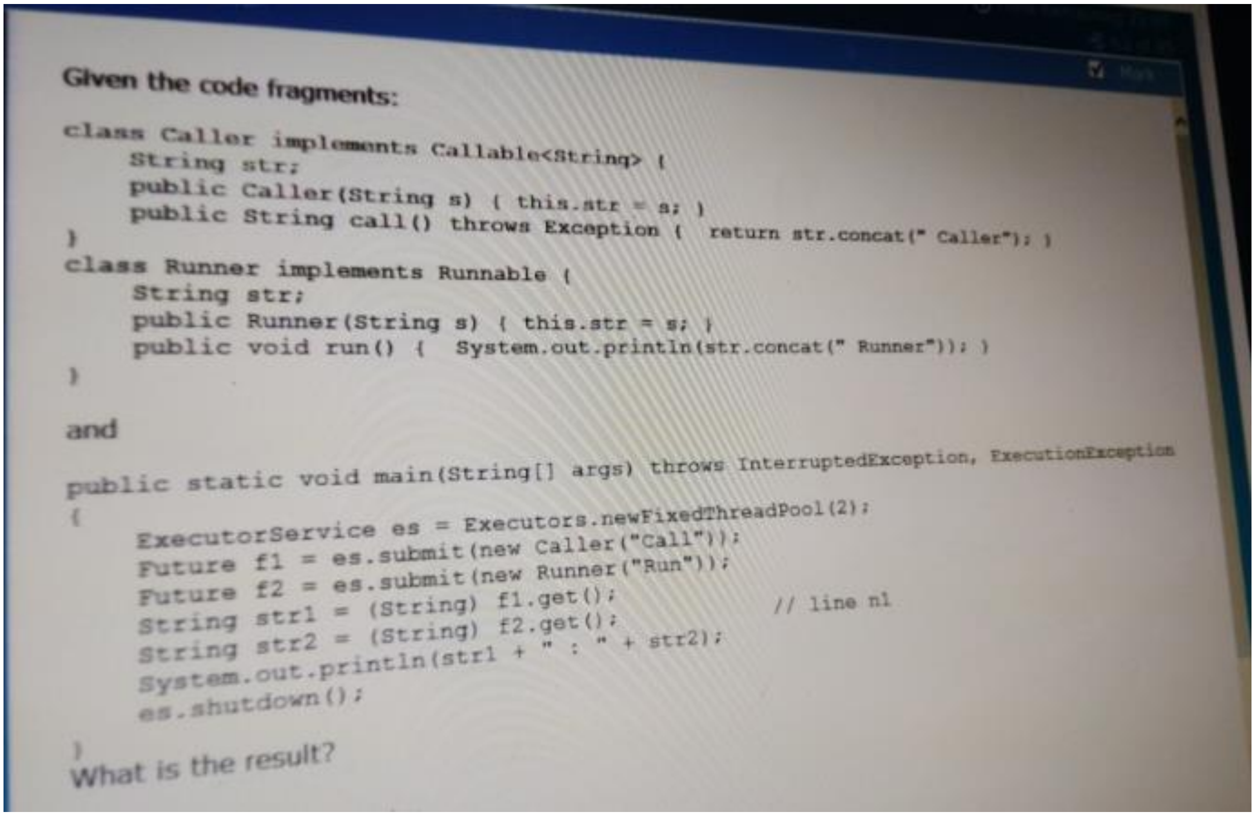
 * Answer: C

 */

public class Product {

```
    String name;
    Integer price;

    Product(String name, int price) {
        this.name = name;
        this.price = price;
    }

    public void printVal() {
        System.out.print(name + " Price: " + price + " ");
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public Integer getPrice() {
        return price;
    }

    public static void main(String[] args) {
        List<Product> li = Arrays.asList(new Product("TV", 1000), new Product("Refrigerator", 2000));
        Consumer<Product> raise = (Product e) -> e.setPrice(e.getPrice() + 100);
        li.forEach(raise).forEach(Product::printVal);
    }
}
```

Problem 63:



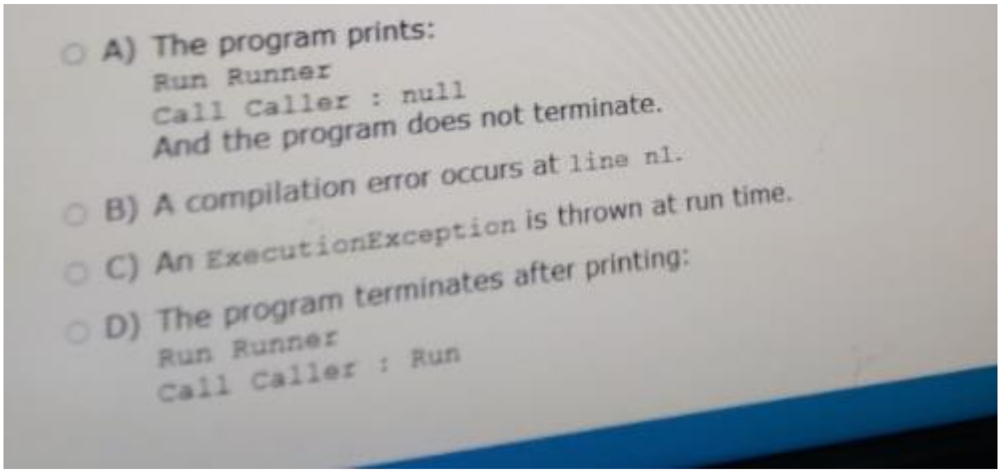
Given the code fragments:

```
class Caller implements Callable<String> {
     String str;
     public Caller(String s) { this.str = s; }
     public String call() throws Exception {  return str.concat(" Caller"); }
}
class Runner implements Runnable {
     String str;
     public Runner(String s) { this.str = s; }
     public void run() {  System.out.println(str.concat(" Runner")); }
}
```

and

```
public static void main(String[] args) throws InterruptedException, ExecutionException
{
     ExecutorService es = Executors.newFixedThreadPool(2);
     Future f1 = es.submit(new Caller("Call"));
     Future f2 = es.submit(new Runner("Run"));
     String str1 = (String) f1.get();                 // line n1
     String str2 = (String) f2.get();
     System.out.println(str1 + " : " + str2);
     es.shutdown();
}
```

What is the result?



○ A) The program prints:

```
Run Runner
Call Caller : null
```

And the program does not terminate.

○ B) A compilation error occurs at `line n1`.

○ C) An `ExecutionException` is thrown at run time.

○ D) The program terminates after printing:

```
Run Runner
Call Caller : Run
```

Correct Answer: A

package com.coderbd.q63;

import java.util.concurrent.*;

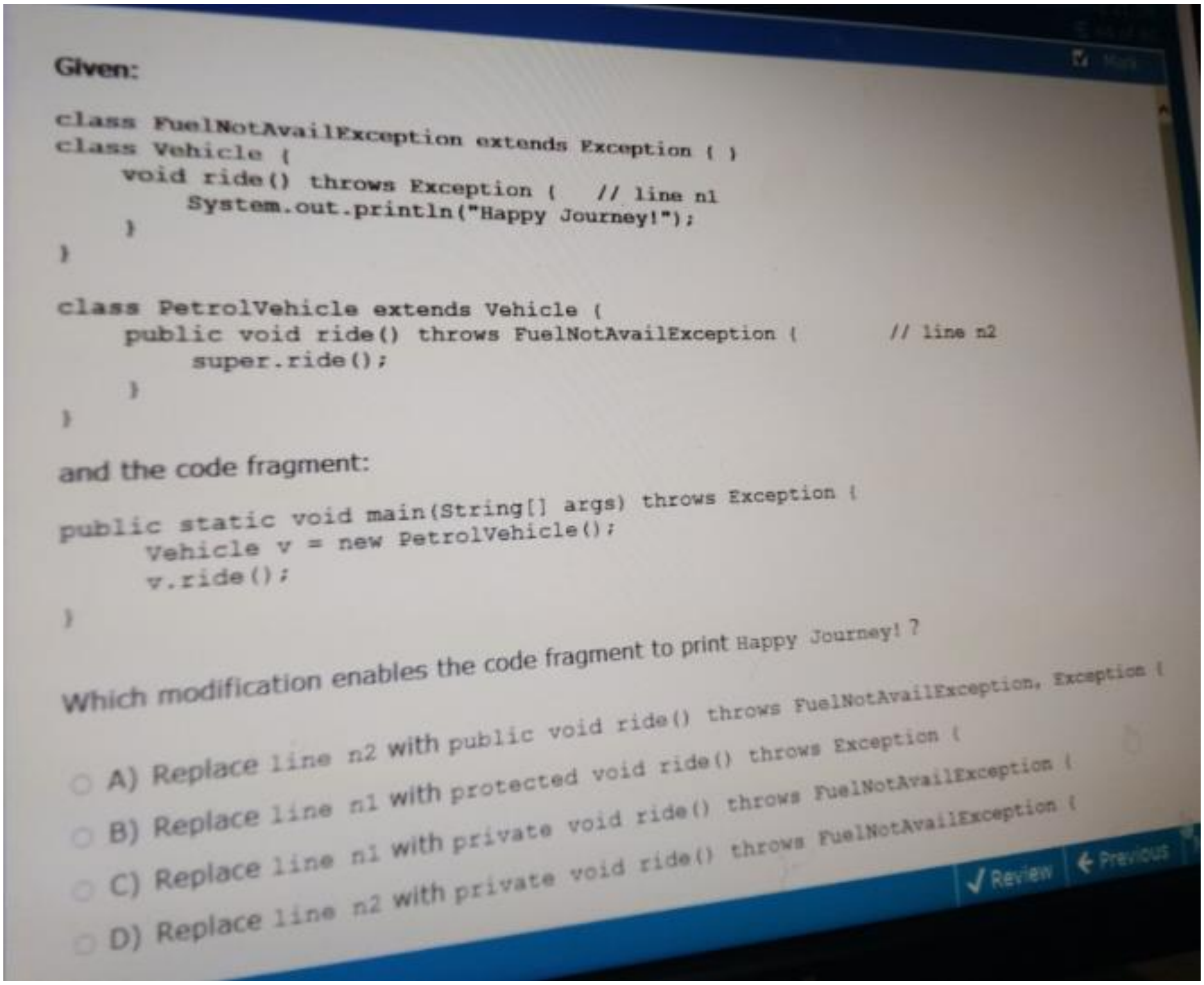
/**

 * Answer: A

```
 */
class Caller implements Callable<String> {
   String str;

   public Caller(String s) {
      this.str = s;
   }

   public String call() throws Exception {
      return str.concat(" Caller");
   }
}

class Runner implements Runnable {
   String str;

   public Runner(String s) {
      this.str = s;
   }

   public void run() {
      System.out.println(str.concat(" Runner"));
   }
}

public class Test {
   public static void main(String[] args) throws InterruptedException, ExecutionException {
      ExecutorService es = Executors.newFixedThreadPool(2);
      Future f1 = es.submit(new Caller("Call"));
```

```
        Future f2 = es.submit(new Runner("Run"));
        String str1 = (String) f1.get();
        String str2 = (String) f2.get();
        System.out.println(str1 + " : " + str2);
        es.shutdown();
    }
}
```

Problem 64:



Given:

```
class FuelNotAvailException extends Exception { }
class Vehicle {
    void ride() throws Exception {   // line n1
        System.out.println("Happy Journey!");
    }
}

class PetrolVehicle extends Vehicle {
    public void ride() throws FuelNotAvailException {        // line n2
        super.ride();
    }
}
```

and the code fragment:

```
public static void main(String[] args) throws Exception {
    Vehicle v = new PetrolVehicle();
    v.ride();
}
```

Which modification enables the code fragment to print Happy Journey! ?

○ A) Replace line n2 with public void ride() throws FuelNotAvailException, Exception {

○ B) Replace line n1 with protected void ride() throws Exception {

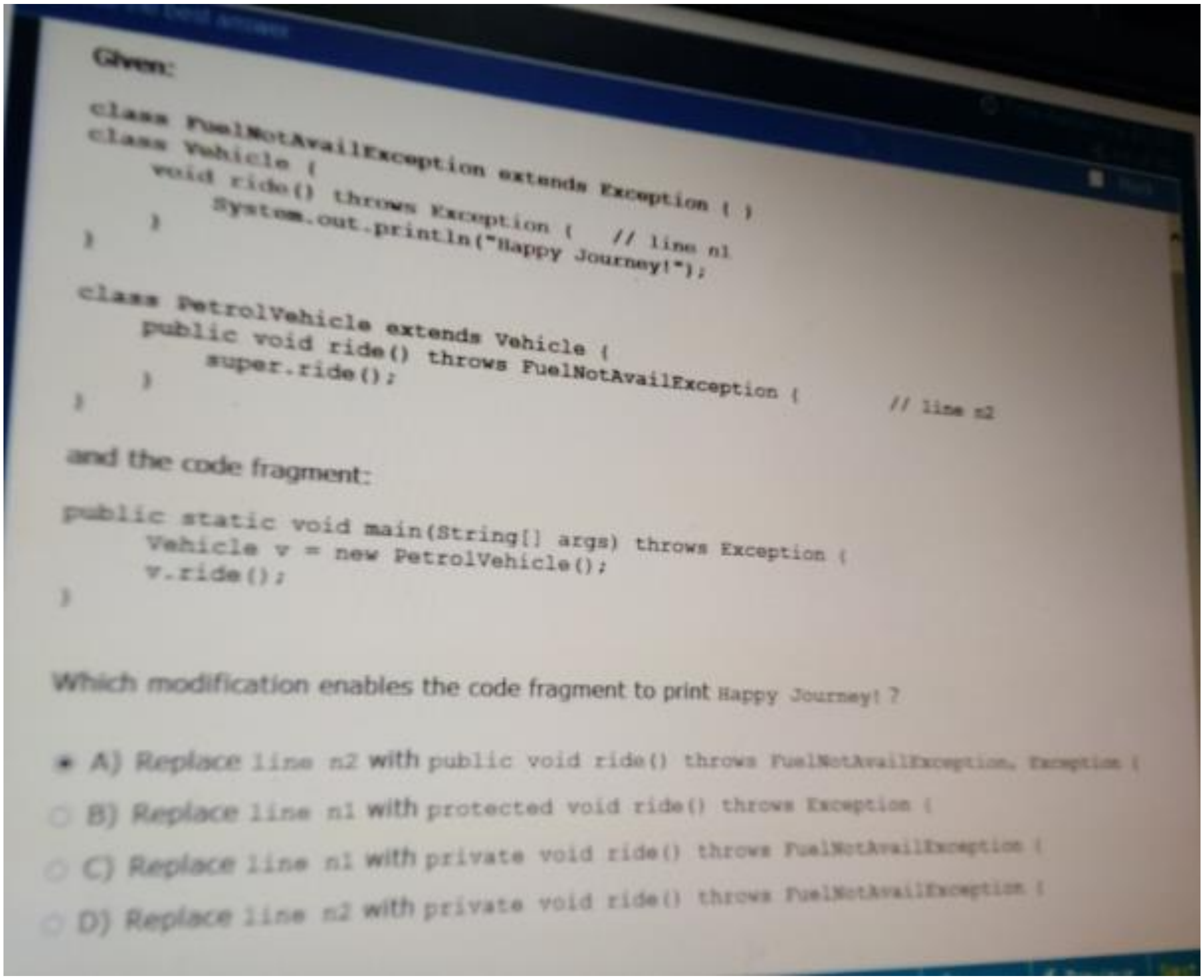
○ C) Replace line n1 with private void ride() throws FuelNotAvailException {

○ D) Replace line n2 with private void ride() throws FuelNotAvailException {

Correct Answer: A

```
package com.coderbd.q64;

/**
 * Answer: A
 */
class FuelNotAvailException extends Exception {
}

class Vehicle {
    void ride() throws Exception {
        System.out.println("Happy Journey!");
    }
}

class PetrolVehicle extends Vehicle {
    public void ride() throws FuelNotAvailException, Exception {
        super.ride();
    }
}

public class Test {
    public static void main(String[] args) throws Exception {
        Vehicle v = new PetrolVehicle();
        v.ride();
    }
}
```

Problem 65:



Given:

```
class FuelNotAvailException extends Exception { }
class Vehicle {
    void ride() throws Exception {    // line n1
        System.out.println("Happy Journey!");
    }
}

class PetrolVehicle extends Vehicle {
    public void ride() throws FuelNotAvailException {        // line n2
        super.ride();
    }
}
```

and the code fragment:

```
public static void main(String[] args) throws Exception {
    Vehicle v = new PetrolVehicle();
    v.ride();
}
```

Which modification enables the code fragment to print `Happy Journey!` ?

● A) Replace `line n2` with `public void ride() throws FuelNotAvailException, Exception {`

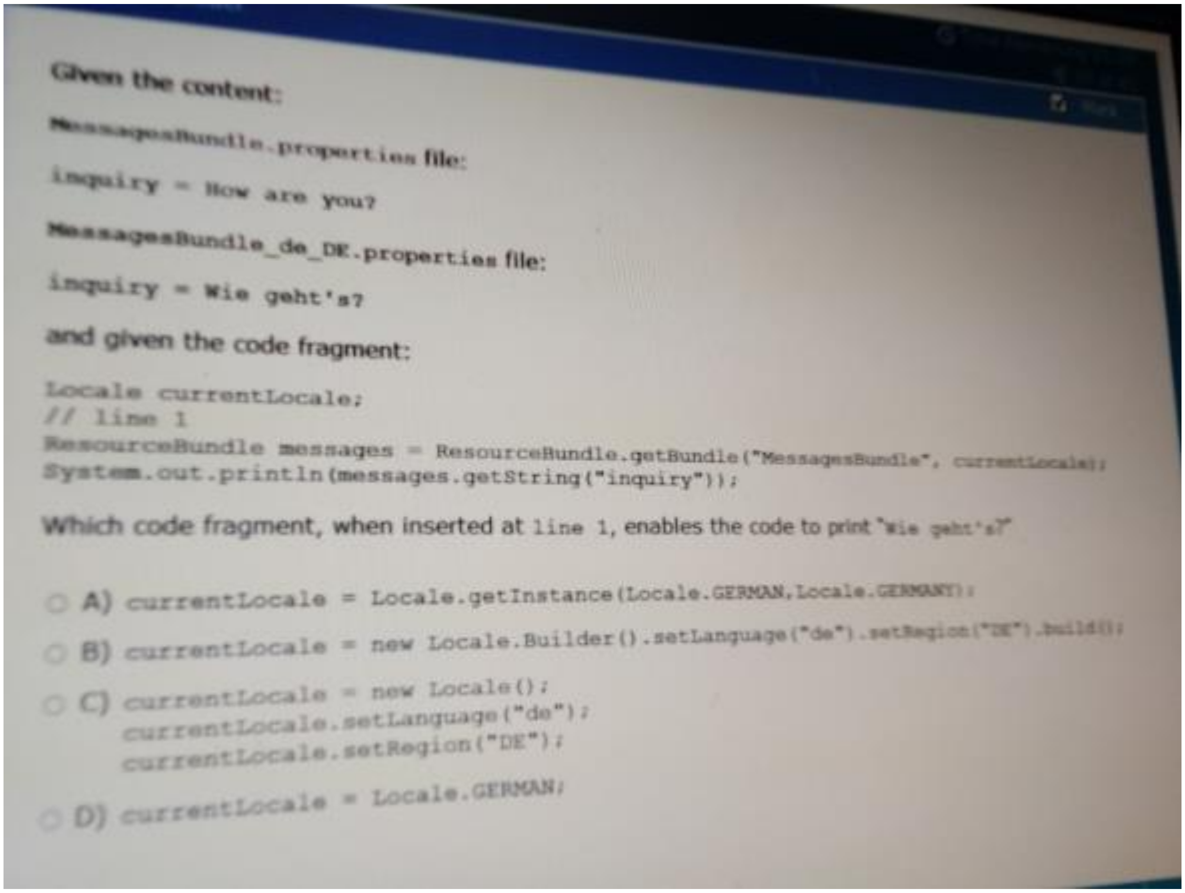
○ B) Replace `line n1` with `protected void ride() throws Exception {`

○ C) Replace `line n1` with `private void ride() throws FuelNotAvailException {`

○ D) Replace `line n2` with `private void ride() throws FuelNotAvailException {`

```
package com.coderbd.q65;


import java.util.Locale;
import java.util.ResourceBundle;


/**
 * Answer: B
 */
public class Test {
   public static void main(String[] args) {
        Locale currentLocale;
        currentLocale = new Locale.Builder().setLanguage("de").setRegion("DE").build();
        ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale);
        System.out.println(messages.getString("inquiry"));
    }
}
```

Problem 66:



Given the content:

`MessagesBundle.properties` file:

```
inquiry = How are you?
```

`MessagesBundle_de_DE.properties` file:

```
inquiry = Wie geht's?
```

and given the code fragment:

```
Locale currentLocale;
// line 1
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
System.out.println(messages.getString("inquiry"));
```

Which code fragment, when inserted at `line 1`, enables the code to print "`Wie geht's?`"

○ A) `currentLocale = Locale.getInstance(Locale.GERMAN,Locale.GERMANY);`

○ B) `currentLocale = new Locale.Builder().setLanguage("de").setRegion("DE").build();`

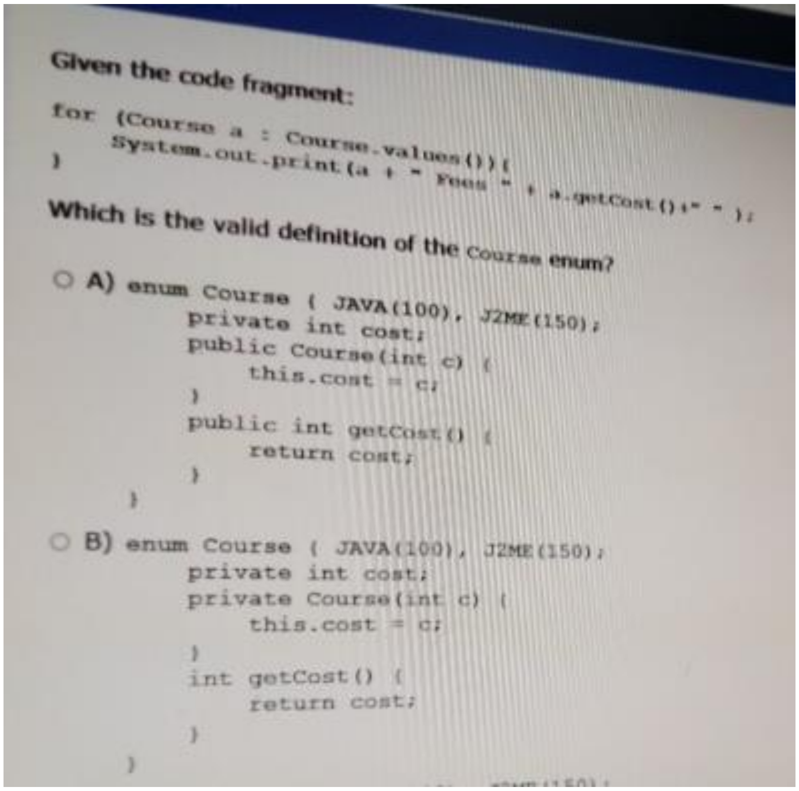
○ C)
```
currentLocale = new Locale();
currentLocale.setLanguage("de");
currentLocale.setRegion("DE");
```

○ D) `currentLocale = Locale.GERMAN;`

```
package com.coderbd.q66;

import java.util.Locale;
import java.util.ResourceBundle;

/**
```

```
 * Answer: B
 */
public class ResourceApp {
    public void loadResourceBundle(){
        ResourceBundle resource = ResourceBundle.getBundle("Greetings", Locale.US);
        System.out.println(resource.getString("hello_msg"));
    }

    public static void main(String[] args) {
        new ResourceApp().loadResourceBundle();
    }
}
```

Problem 67:



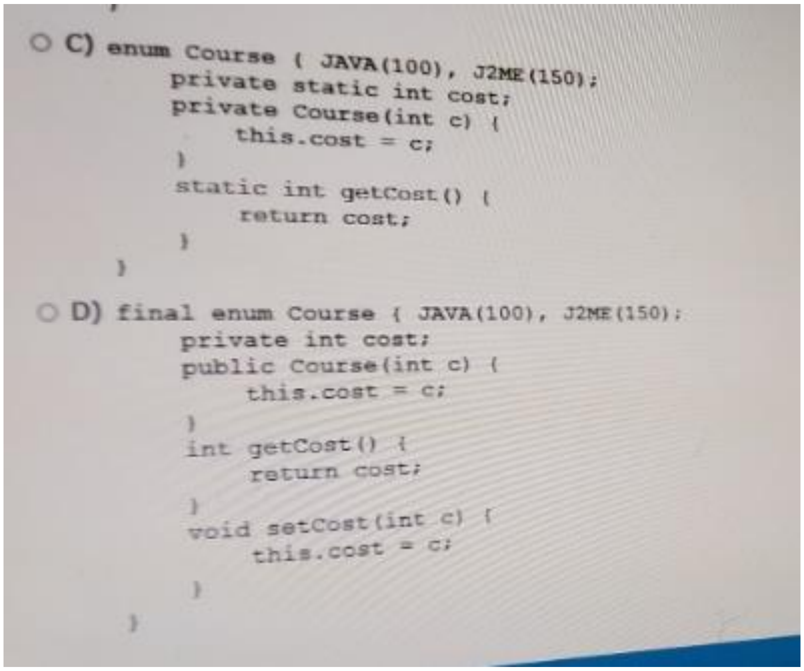
Given the code fragment:

```
for (Course a : Course.values()){
    System.out.print(a + " Fees " + a.getCost()+" " );
}
```

Which is the valid definition of the Course enum?

○ A)
```
enum Course { JAVA(100), J2ME(150);
        private int cost;
        public Course(int c) {
            this.cost = c;
        }
        public int getCost() {
            return cost;
        }
    }
```

○ B)
```
enum Course { JAVA(100), J2ME(150);
        private int cost;
        private Course(int c) {
            this.cost = c;
        }
        int getCost() {
            return cost;
        }
    }
```



○ C)
```
enum Course { JAVA(100), J2ME(150);
        private static int cost;
        private Course(int c) {
            this.cost = c;
        }
        static int getCost() {
            return cost;
        }
    }
```

○ D)
```
final enum Course { JAVA(100), J2ME(150);
        private int cost;
        public Course(int c) {
            this.cost = c;
        }
        int getCost() {
            return cost;
        }
        void setCost(int c) {
            this.cost = c;
        }
    }
```

Correct Answer: B

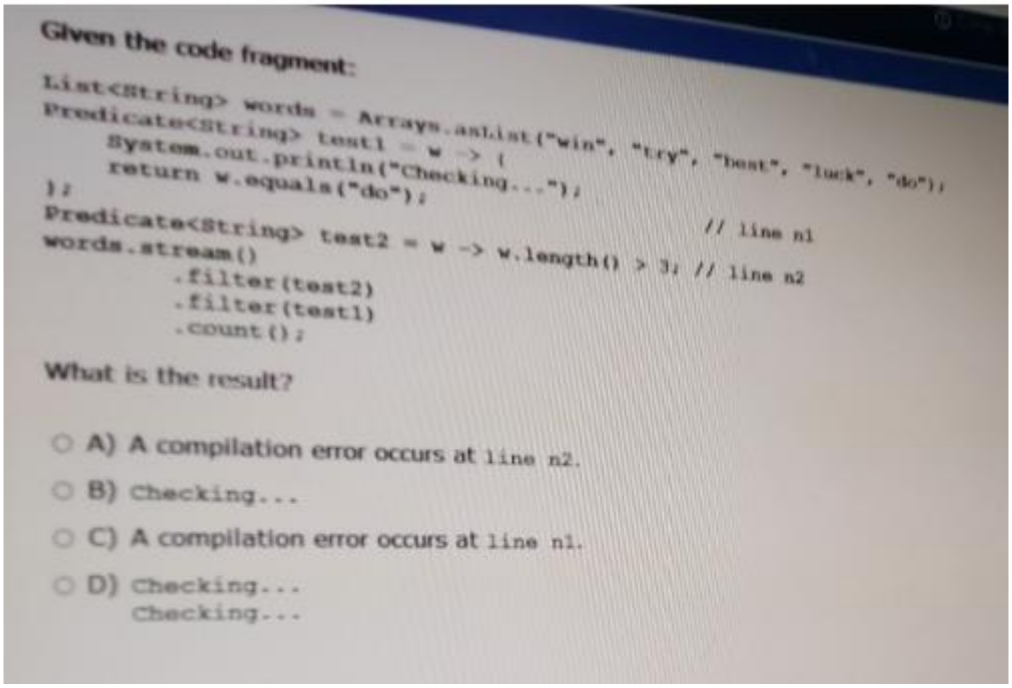
package com.coderbd.q67;

/**

```
 * Answer B
 */

enum Course {
    JAVA(100), J2ME(150);
    private int cost;

    private Course(int c) {
        this.cost = c;
    }

    public int getCost() {
        return cost;
    }
}

public class Test {
    public static void main(String[] args) {
        for (Course a : Course.values()) {
            System.out.print(a + " Fees " + a.getCost() + " ");
        }
    }
}
```

Problem 68:

;



Given the code fragment:

```
List<String> words = Arrays.asList("win", "try", "best", "luck", "do");
Predicate<String> test1 = w -> {
    System.out.println("Checking...");
    return w.equals("do");
};                                              // line n1
Predicate<String> test2 = w -> w.length() > 3; // line n2
words.stream()
        .filter(test2)
        .filter(test1)
        .count();
```

What is the result?

○ A) A compilation error occurs at line n2.

○ B) Checking...

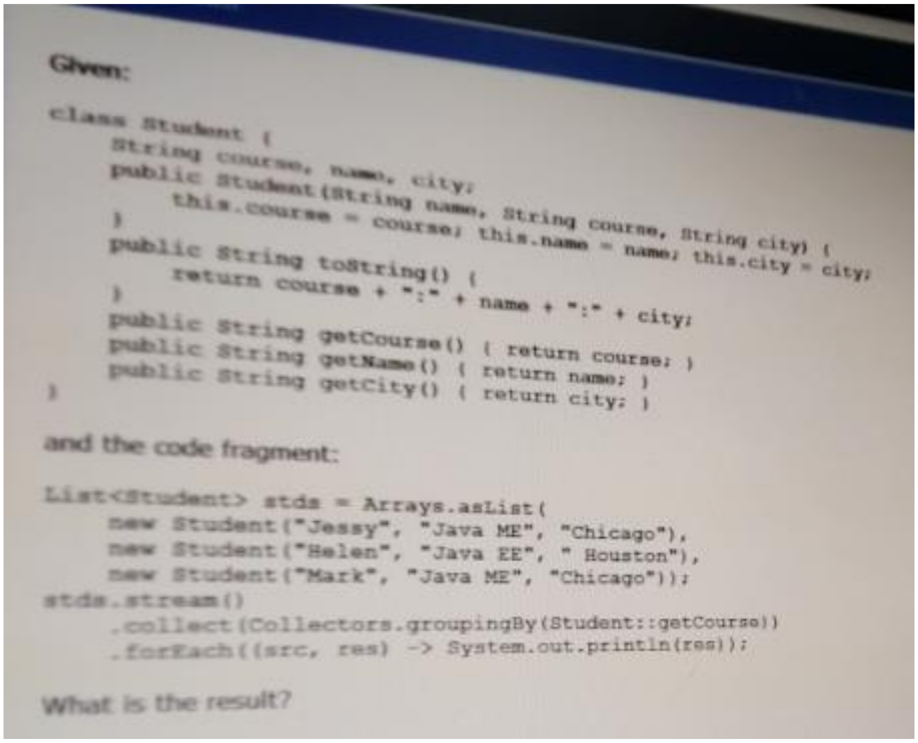
○ C) A compilation error occurs at line n1.

○ D) Checking...
Checking...

Correct Answer. D

```
package com.coderbd.q68;


import java.util.Arrays;

import java.util.List;

import java.util.function.Predicate;


/**

 * Answer: D

 */

public class Test {

    public static void main(String[] args) {

        List<String> words = Arrays.asList("win", "try", "best", "luck", "do");

        Predicate<String> test1 = w -> {

            System.out.println("Checking...");

            return w.equals("do");

        };
    Predicate<String> test2 = w -> w.length() > 3;
    words.stream()
        .filter(test2)
        .filter(test1)
        .count();
  }
}
```

Problem 69:



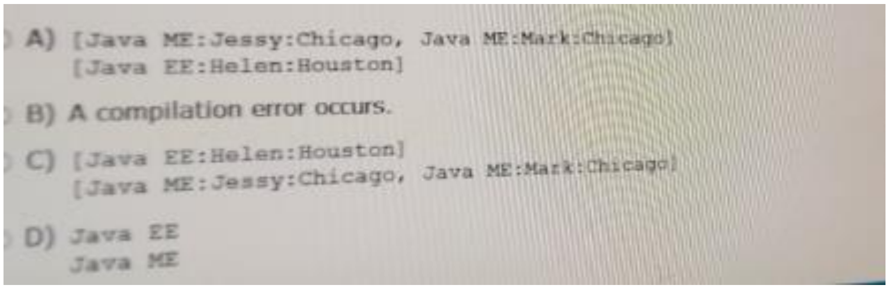
Given:

```
class Student {
    String course, name, city;
    public Student(String name, String course, String city) {
        this.course = course; this.name = name; this.city = city;
    }
    public String toString() {
        return course + ":" + name + ":" + city;
    }
    public String getCourse() { return course; }
    public String getName() { return name; }
    public String getCity() { return city; }
}
```

and the code fragment:

```
List<Student> stds = Arrays.asList(
    new Student("Jessy", "Java ME", "Chicago"),
    new Student("Helen", "Java EE", " Houston"),
    new Student("Mark", "Java ME", "Chicago"));
stds.stream()
    .collect(Collectors.groupingBy(Student::getCourse))
    .forEach((src, res) -> System.out.println(res));
```

What is the result?



A) [Java ME:Jessy:Chicago, Java ME:Mark:Chicago]
[Java EE:Helen:Houston]

B) A compilation error occurs.

C) [Java EE:Helen:Houston]
[Java ME:Jessy:Chicago, Java ME:Mark:Chicago]

D) Java EE
Java ME

Correct Answer: C

package com.coderbd.q69;

import java.util.Arrays;

import java.util.List;

import java.util.stream.Collectors;

/**

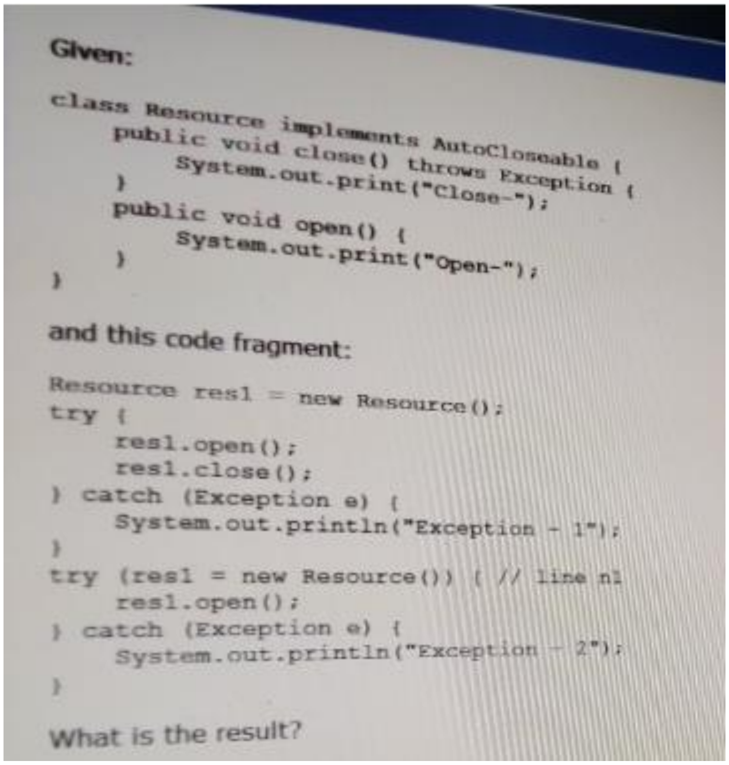
 * Answer: C

 */

class Student {

```
    String course, name, city;

    public Student(String name, String course, String city) {
        this.course = course;
        this.name = name;
        this.city = city;
    }

    public String toString() {
        return course + ":" + name + ":" + city;
    }

    public String getCourse() {
        return course;
    }

    public String getName() {
        return name;
    }

    public String getCity() {
        return city;
    }
}

public class Test {
    public static void main(String[] args) {
        List<Student> stds = Arrays.asList(
                new Student("Jessy", "Java ME", "Chicago"),
            new Student("Helen", "Java EE", " Houston"),
            new Student("Mark", "Java ME", "Chicago"));
        stds.stream()
            .collect(Collectors.groupingBy(Student::getCourse))
            .forEach((src, res) -> System.out.println(res));
    }
}
```

Problem 70:



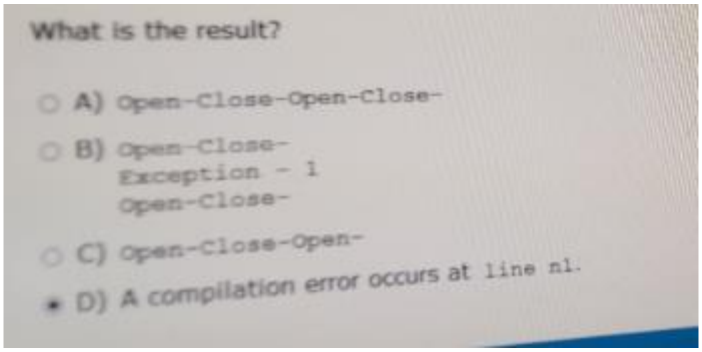
Given:

```
class Resource implements AutoCloseable {
    public void close() throws Exception {
        System.out.print("Close-");
    }
    public void open() {
        System.out.print("Open-");
    }
}
```

and this code fragment:

```
Resource res1 = new Resource();
try {
    res1.open();
    res1.close();
} catch (Exception e) {
    System.out.println("Exception - 1");
}
try (res1 = new Resource()) { // line n1
    res1.open();
} catch (Exception e) {
    System.out.println("Exception - 2");
}
```

What is the result?



What is the result?

○ A) Open-Close-Open-Close-

○ B) Open-Close-
Exception - 1
Open-Close-

○ C) Open-Close-Open-

● D) A compilation error occurs at line n1.

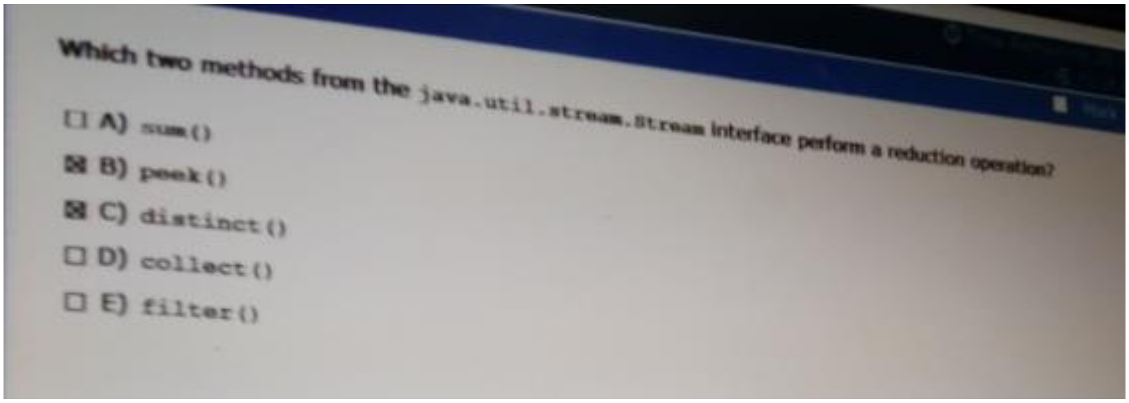
Correct Answer: C

package com.coderbd.q70;

```
/**
 * Answer: C
 */
class Resource implements AutoCloseable {
    public void close() throws Exception {
        System.out.print("Close-");
    }

    public void open() {
        System.out.print("Open-");
    }
}

public class Test {
    public static void main(String[] args) {
        Resource res1 = new Resource();
        try {
            res1.open();
            res1.close();
        } catch (Exception e) {
            System.out.println("Exception - 1");
        }
        try {
            res1.open();
        } catch (Exception e) {
            System.out.println("Exception - 2");
        }
    }
```

}

Problem 71:



Which two methods from the java.util.stream.Stream interface perform a reduction operation?

□ A) sum()

☒ B) peek()

☒ C) distinct()

□ D) collect()

□ E) filter()

Correct Answer: A, D

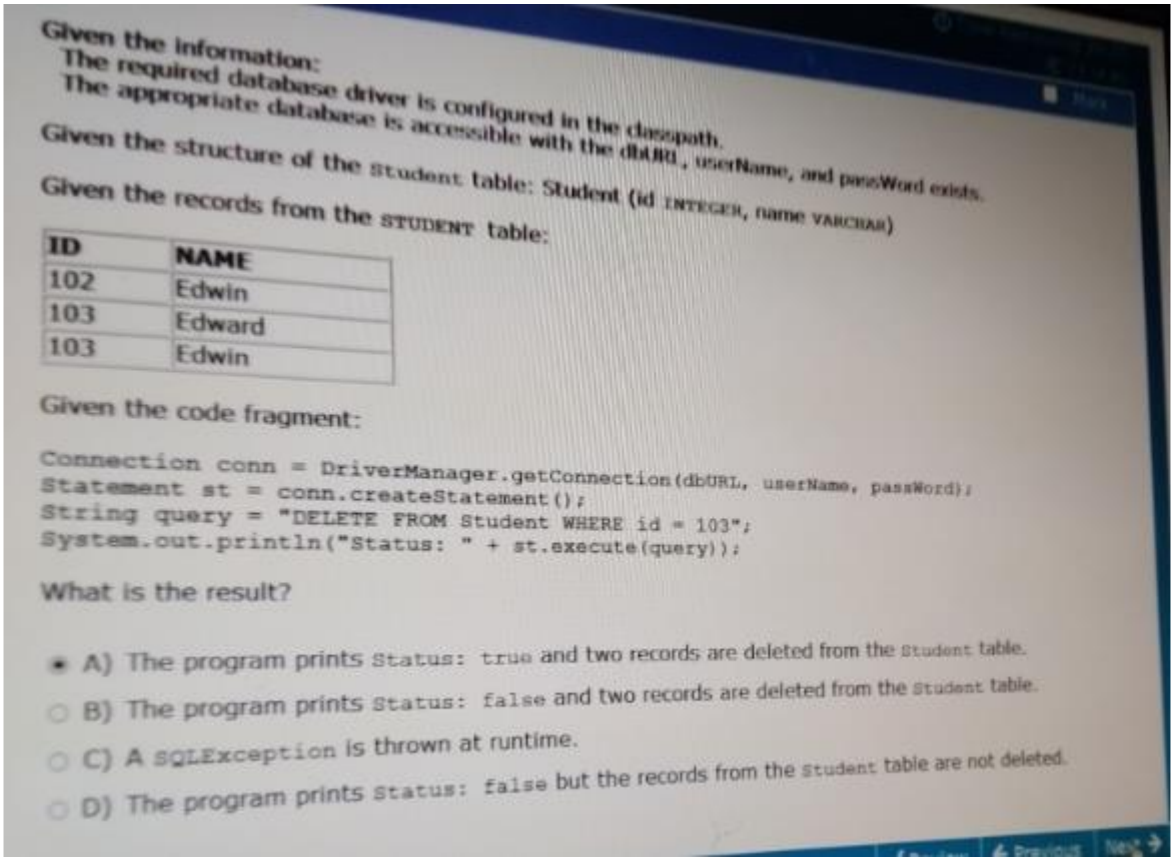
package com.coderbd.Q71;

public class Theory71 {

  // Ans: A,B

}

Problem 72:



Given the information:
The required database driver is configured in the classpath.
The appropriate database is accessible with the dbURL, userName, and passWord exists.

Given the structure of the Student table: Student (id INTEGER, name VARCHAR)

Given the records from the STUDENT table:

ID	NAME
102	Edwin
103	Edward
103	Edwin

Given the code fragment:

```
Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
Statement st = conn.createStatement();
String query = "DELETE FROM Student WHERE id = 103";
System.out.println("Status: " + st.execute(query));
```

What is the result?

● A) The program prints Status: true and two records are deleted from the Student table.

○ B) The program prints Status: false and two records are deleted from the Student table.

○ C) A SQLException is thrown at runtime.

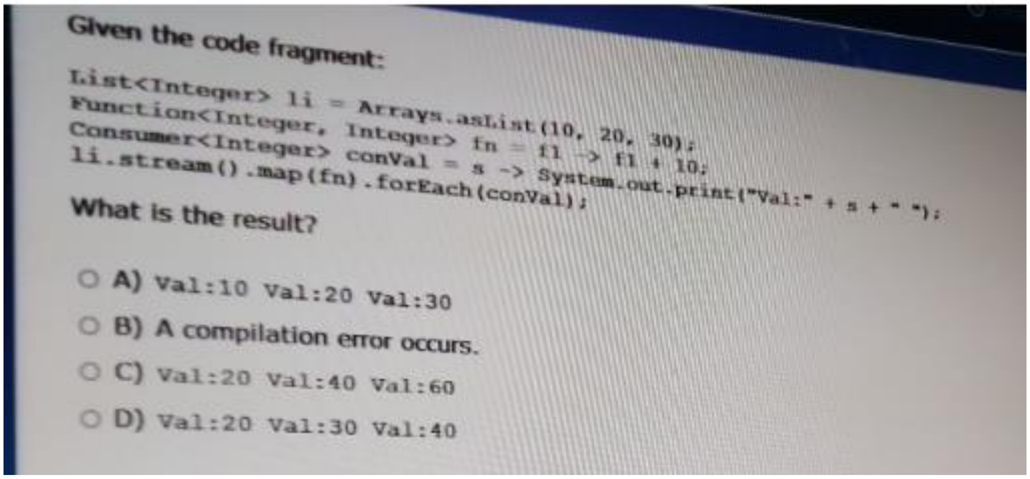
○ D) The program prints Status: false but the records from the Student table are not deleted.

Correct Answer: D

```
package com.coderbd.Q72;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;


/**
```

```
 *
 * @author Touhid
 */
public class Test {
   static String dbURL;
   static String passWord;
   static String userName;
   public static void main(String[] args) throws SQLException {
      Connection conn=DriverManager.getConnection(dbURL,userName,passWord);
      Statement st=conn.createStatement();
      String query="DELETE FROM Student WHERE id=103";
      System.out.println("Status: "+st.execute(query));
   }
   ////Ans:B
}
```

Problem 73:



Given the code fragment:

```
List<Integer> li = Arrays.asList(10, 20, 30);
Function<Integer, Integer> fn = f1 -> f1 + 10;
Consumer<Integer> conVal = s -> System.out.print("Val:" + s + " ");
li.stream().map(fn).forEach(conVal);
```

What is the result?

○ A) Val:10 Val:20 Val:30

○ B) A compilation error occurs.

○ C) Val:20 Val:40 Val:60

○ D) Val:20 Val:30 Val:40

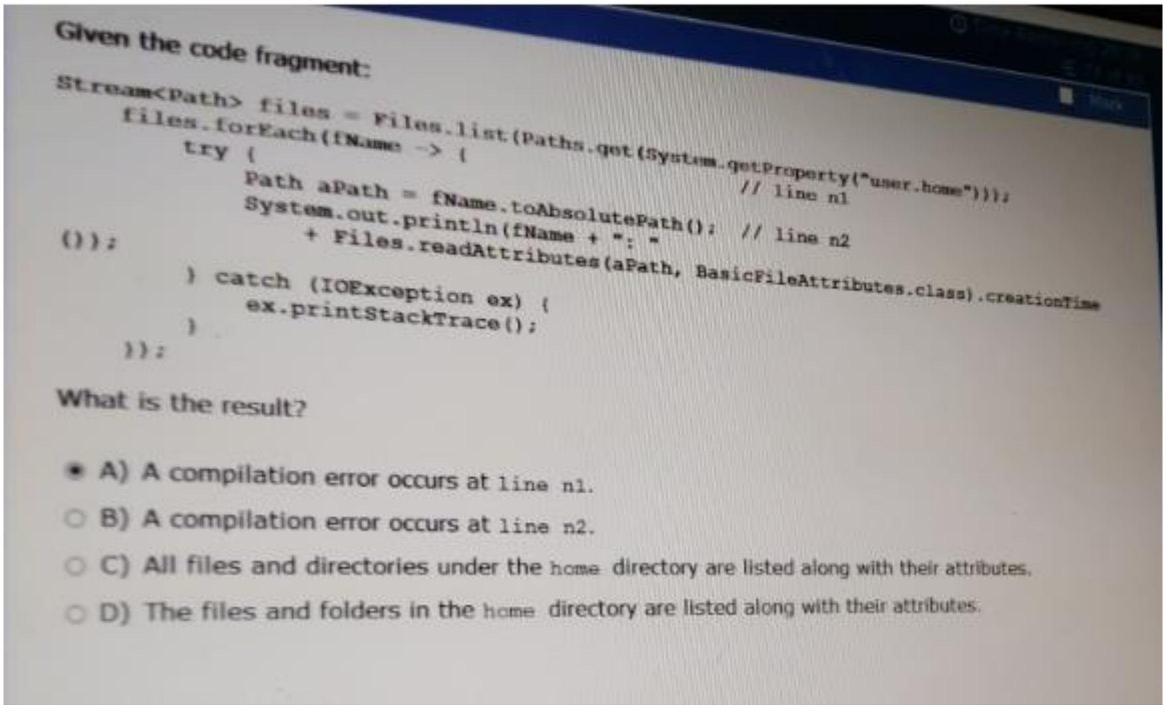
Correct Answer: D

package com.coderbd.Q73;

import java.util.Arrays;

```
import java.util.List;

import java.util.function.Consumer;

import java.util.function.Function;


public class Test {

    public static void main(String[] args) {

        List<Integer> li=Arrays.asList(10,20,30);

        Function<Integer, Integer> fn= f1 -> f1+10;

        Consumer<Integer> conVal= s -> System.out.print("Val:"+s+" ");

         li.stream().map(fn).forEach(conVal);


    }

    ///Ans:D

}
```

Problem 74:



Given the code fragment:

```
Stream<Path> files = Files.list(Paths.get(System.getProperty("user.home")));
     files.forEach(fName -> {                                  // line n1
         try {
              Path aPath = fName.toAbsolutePath();   // line n2
              System.out.println(fName + ": "
                   + Files.readAttributes(aPath, BasicFileAttributes.class).creationTime
());
         } catch (IOException ex) {
              ex.printStackTrace();
         }
     });
```

What is the result?

● A) A compilation error occurs at line n1.

○ B) A compilation error occurs at line n2.

○ C) All files and directories under the home directory are listed along with their attributes.

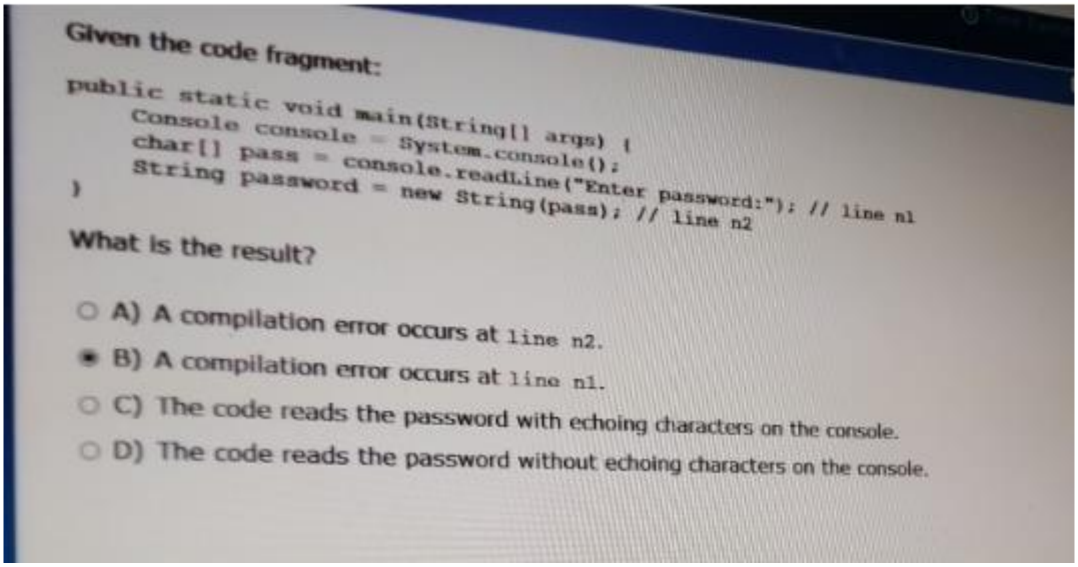
○ D) The files and folders in the home directory are listed along with their attributes.

Correct Answer: C

package com.coderbd.Q74;

```
import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.attribute.BasicFileAttributes;

import java.util.stream.Stream;


public class Test {


    public static void main(String[] args) throws IOException {

        Stream<Path> files = Files.list(Paths.get(System.getProperty("user.home")));

        files.forEach(fName -> {                //line n1

            try {

                Path apath = fName.toAbsolutePath();     //line n2

                System.out.println(fName + ": "

                        + Files.readAttributes(apath, BasicFileAttributes.class).creationTime());

            } catch (IOException ex) {

                   ex.printStackTrace();

            }

        });

               // Ans: Confuse............

    }

}
```

Problem 75:



Given the code fragment:

```
public static void main(String[] args) {
    Console console = System.console();
    char[] pass = console.readLine("Enter password:"); // line n1
    String password = new String(pass); // line n2
}
```

What is the result?

○ A) A compilation error occurs at `line n2`.

● B) A compilation error occurs at `line n1`.

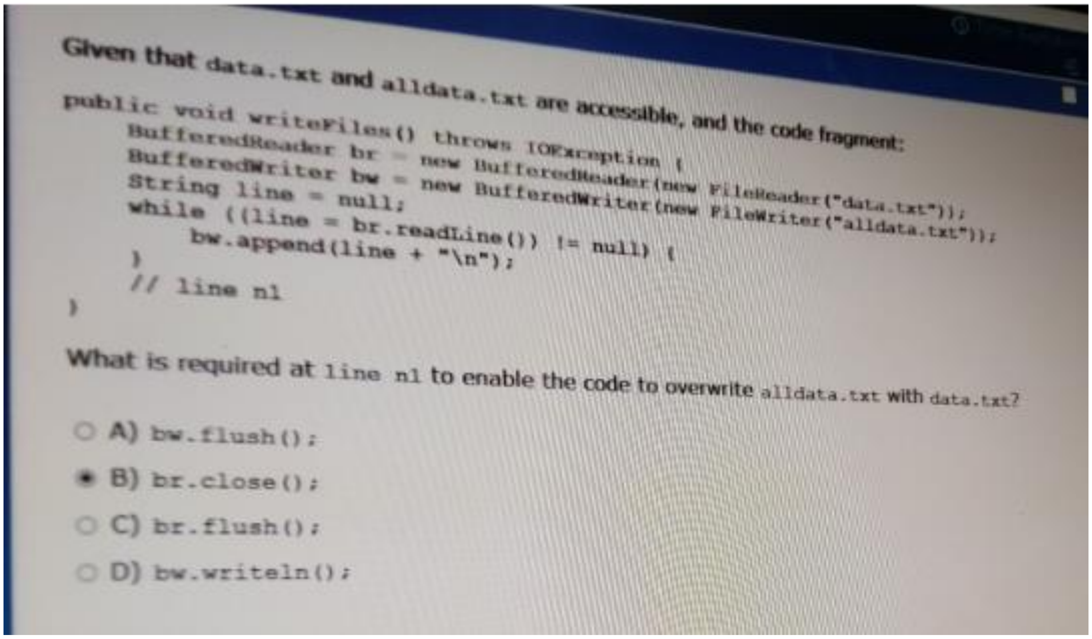
○ C) The code reads the password with echoing characters on the console.

○ D) The code reads the password without echoing characters on the console.

Correct Answer: B

```
package com.coderbd.Q75;


import java.io.Console;


public class Test {
    public static void main(String[] args) {
        Console console = System.console();
        char[] pass =console.readLine("Enter password:");  //line n1
        String password =new String(pass);
        String Password =new String(pass);   //line n2


    }
   // Ans: B
}
```

Problem 76:



Given that `data.txt` and `alldata.txt` are accessible, and the code fragment:

```
public void writeFiles() throws IOException {
    BufferedReader br = new BufferedReader(new FileReader("data.txt"));
    BufferedWriter bw = new BufferedWriter(new FileWriter("alldata.txt"));
    String line = null;
    while ((line = br.readLine()) != null) {
        bw.append(line + "\n");
    }
    // line n1
}
```

What is required at `line n1` to enable the code to overwrite `alldata.txt` with `data.txt`?

○ A) `bw.flush();`

● B) `br.close();`

○ C) `br.flush();`

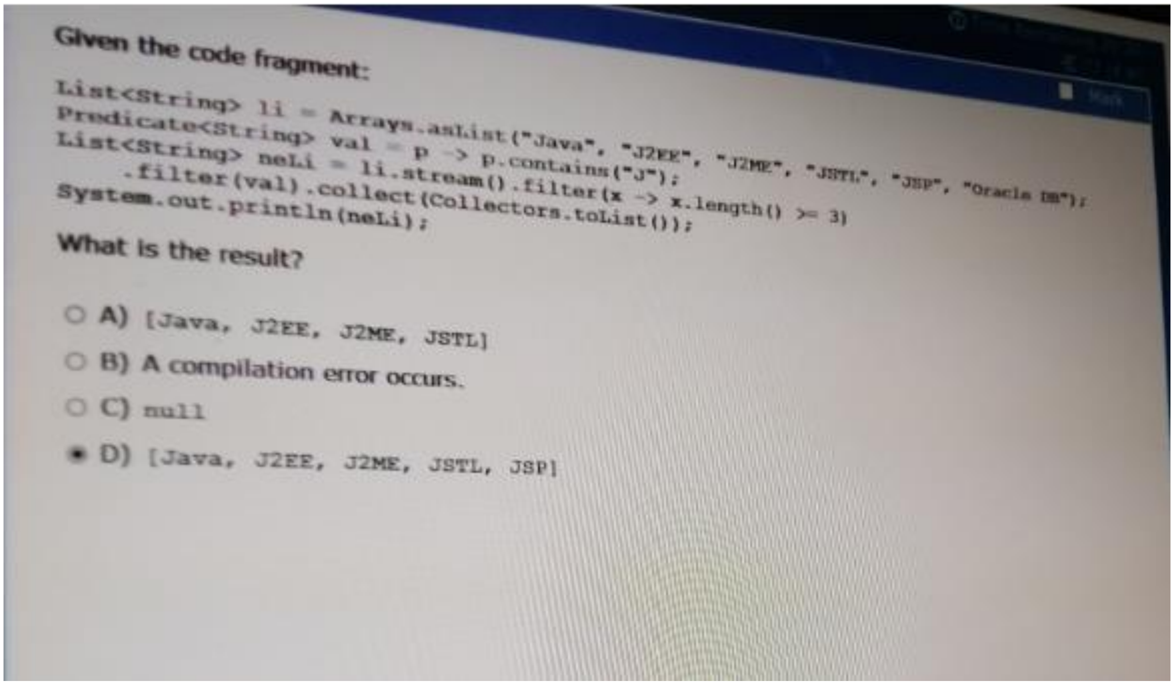
○ D) `bw.writeln();`

Correct Answer: A

```
package com.coderbd.Q76;


import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;


public class Test {

  public static void main(String[] args) throws IOException {

    BufferedReader br=new BufferedReader(new FileReader("data.txt"));

    BufferedWriter bw=new BufferedWriter(new FileWriter("alldata.txt"));

    String line=null;

    while((line=br.readLine()) !=null){

      bw.append(line + "\n");

    }

    //A bw.flush();
```

```
     //B  br.close();
    //C  br.flush();
    //bw.writeln();
      //line n1
   }
}//      Ans: confuse
```

Problem 77:



Given the code fragment:

```
List<String> li = Arrays.asList("Java", "J2EE", "J2ME", "JSTL", "JSP", "Oracle DB");
Predicate<String> val = p -> p.contains("J");
List<String> neLi = li.stream().filter(x -> x.length() >= 3)
     .filter(val).collect(Collectors.toList());
System.out.println(neLi);
```

What is the result?

○ A) [Java, J2EE, J2ME, JSTL]

○ B) A compilation error occurs.

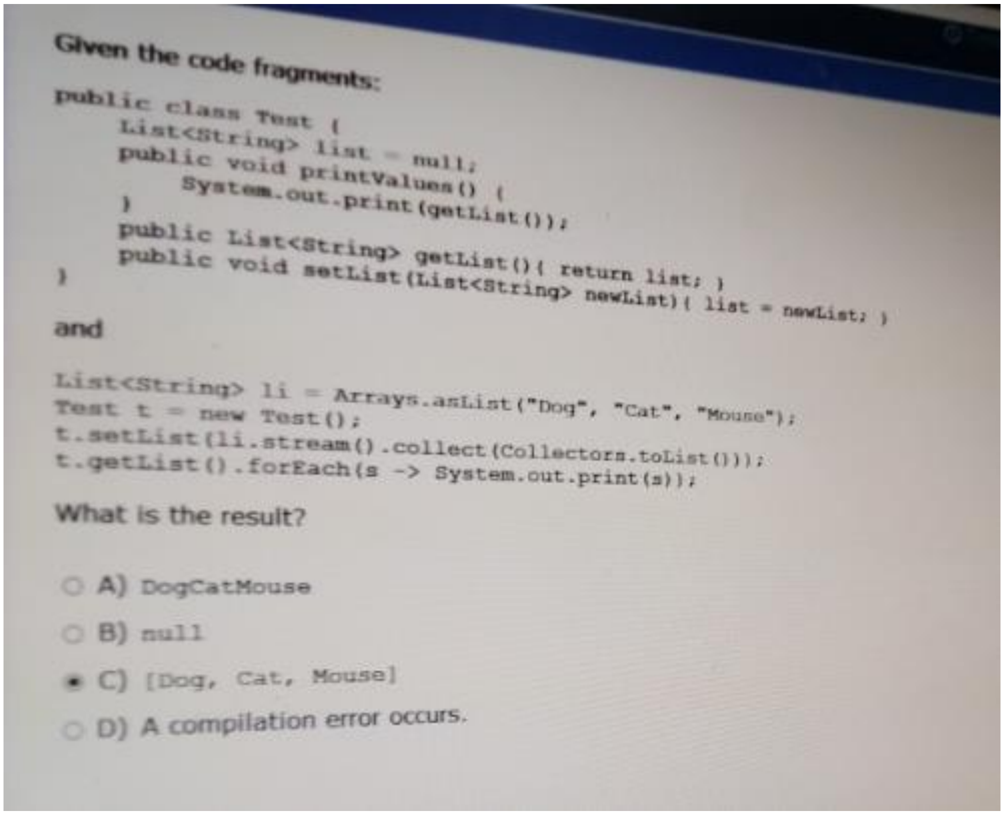
○ C) null

● D) [Java, J2EE, J2ME, JSTL, JSP]

Correct Answer: D

```
package com.coderbd.Q77;

import java.util.Arrays;
import java.util.List;
import java.util.function.Predicate;
import java.util.stream.Collectors;

public class Test {
    public static void main(String[] args) {
        List <String> li=Arrays.asList("Java","J2EE","J2ME","JSTL","JSP","Oracle DB");
        Predicate<String> val = p -> p.contains("J");
        List<String> neli=li.stream().filter(x -> x.length() >= 3)
                .filter(val).collect(Collectors.toList());
        System.out.println(neli);
    }
    //// Ans: D
}
```

Problem 78:



Given the code fragments:

```
public class Test {
    List<String> list = null;
    public void printValues() {
        System.out.print(getList());
    }
    public List<String> getList(){ return list; }
    public void setList(List<String> newList){ list = newList; }
}
```

and

```
List<String> li = Arrays.asList("Dog", "Cat", "Mouse");
Test t = new Test();
t.setList(li.stream().collect(Collectors.toList()));
t.getList().forEach(s -> System.out.print(s));
```

What is the result?

○ A) DogCatMouse

○ B) null

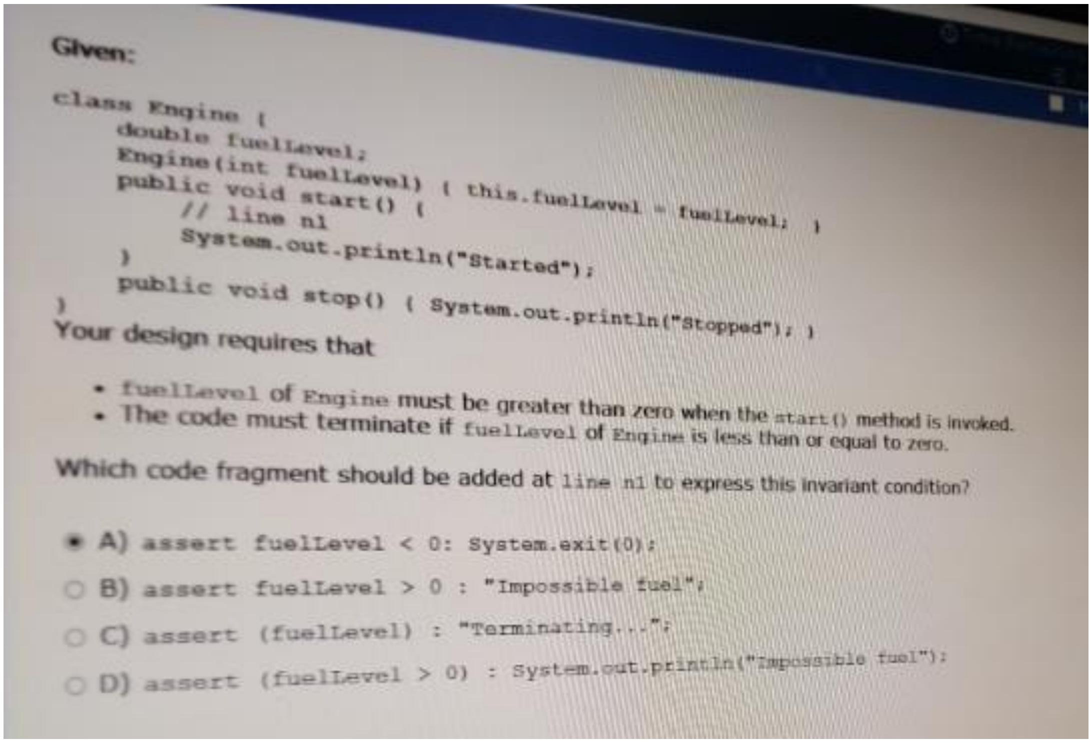
● C) [Dog, Cat, Mouse]

○ D) A compilation error occurs.

Correct Answer: C

```
package com.coderbd.Q78;


import java.util.Arrays;

import java.util.List;

import java.util.stream.Collectors;


public class Test {


   List<String> list = null;


   public void printValues() {

      System.out.println(getList());

   }

   public List<String> getList(){
        return list;
    }
    public void setList(List<String> newList){
        list=newList;
    }

    public static void main(String[] args) {
        List<String> li=Arrays.asList("Dog","Cat","Mouse");
        Test t = new Test();
        t.setList(li.stream().collect(Collectors.toList()));
        t.getList().forEach(s -> System.out.print(s));
    }
    //  Ans: C
}
```

Problem 79:



Given:

```
class Engine {
    double fuelLevel;
    Engine(int fuelLevel) { this.fuelLevel = fuelLevel; }
    public void start() {
        // line n1
        System.out.println("Started");
    }
    public void stop() { System.out.println("Stopped"); }
}
```

Your design requires that

- fuelLevel of Engine must be greater than zero when the start() method is invoked.
- The code must terminate if fuelLevel of Engine is less than or equal to zero.

Which code fragment should be added at line n1 to express this invariant condition?

● A) `assert fuelLevel < 0: System.exit(0);`

○ B) `assert fuelLevel > 0 : "Impossible fuel";`

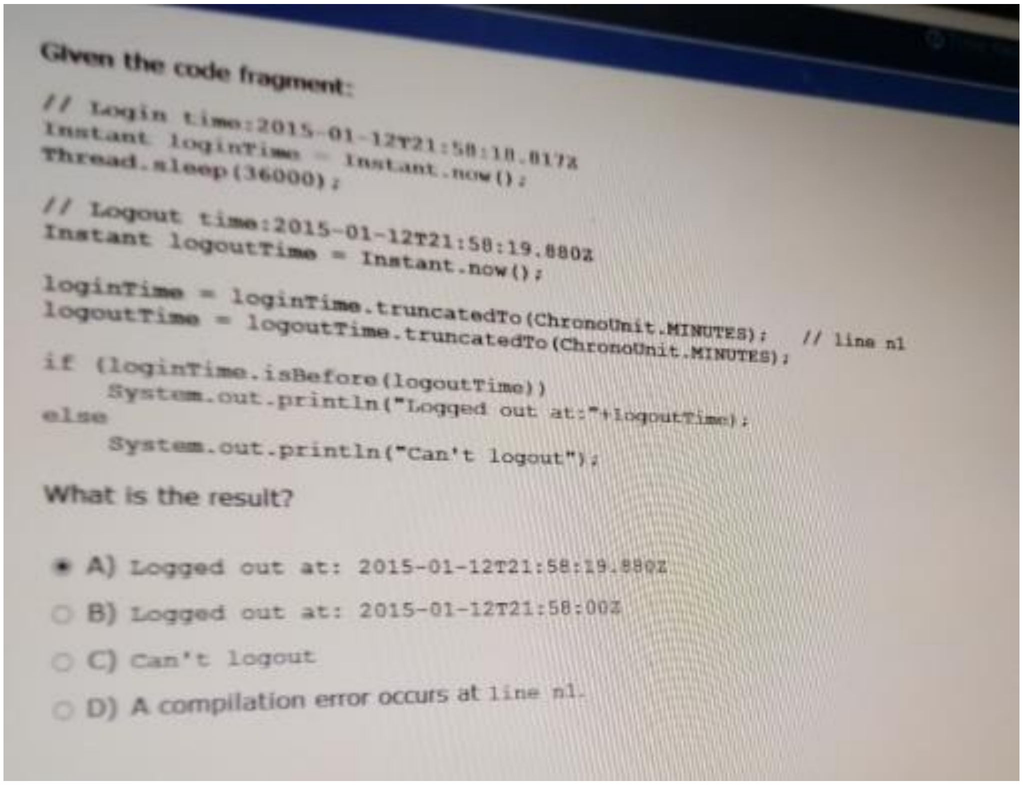
○ C) `assert (fuelLevel) : "Terminating...";`

○ D) `assert (fuelLevel > 0) : System.out.println("Impossible fuel");`

Correct Answer: A, Though this question has problem, but closest answer is A

```
package com.coderbd.Q79;


 class Engine {
    double fuelLevel;
    Engine(int fuelLevel){
       this.fuelLevel=fuelLevel;
    }
    public void start(){
       // Line n1
       assert fuielLevel < 0 : System.exit(0);
       System.out.println("Started");
    }
    public void stop(){
       System.out.println("Stopped");
    }
    public static void main(String[] args) {


    }
   // Ans: Confuse
}
```

Problem 80:



Given the code fragment:

```
// Login time:2015-01-12T21:58:18.817Z
Instant loginTime = Instant.now();
Thread.sleep(36000);

// Logout time:2015-01-12T21:58:19.880Z
Instant logoutTime = Instant.now();

loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES);   // line n1
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);

if (loginTime.isBefore(logoutTime))
    System.out.println("Logged out at:"+logoutTime);
else
    System.out.println("Can't logout");
```

What is the result?

● A) Logged out at: 2015-01-12T21:58:19.880Z

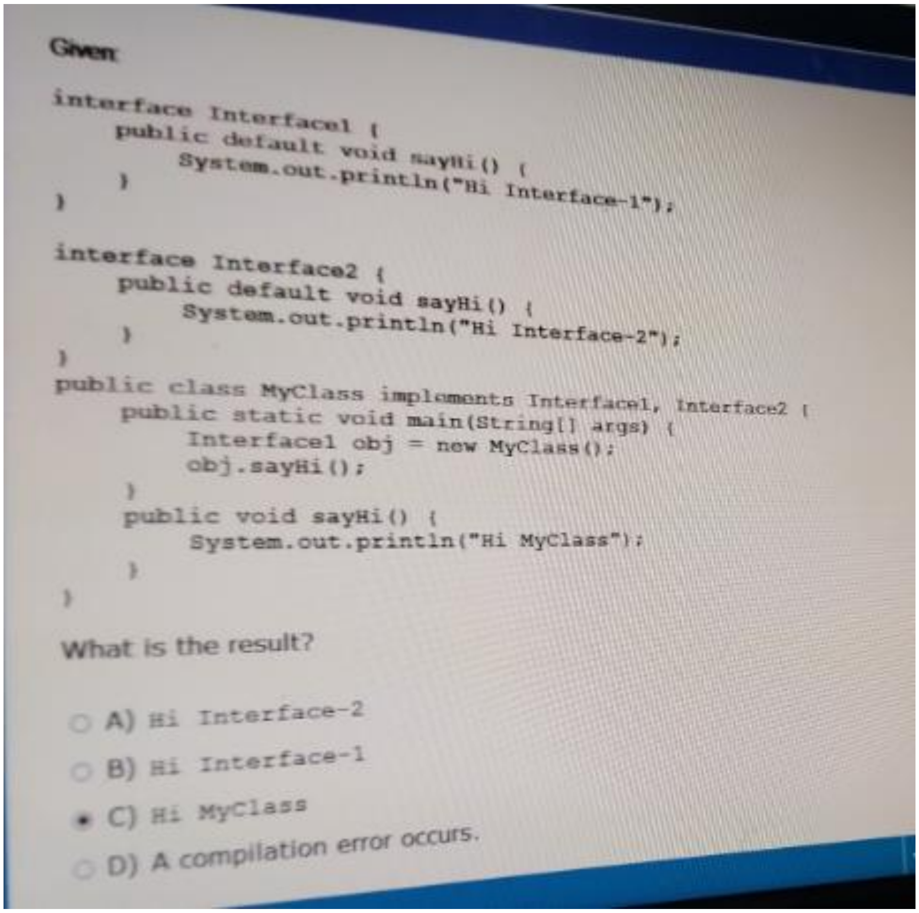
○ B) Logged out at: 2015-01-12T21:58:00Z

○ C) Can't logout

○ D) A compilation error occurs at line n1.

Correct Answer: C

Problem 81:



Given:

```
interface Interface1 {
    public default void sayHi() {
        System.out.println("Hi Interface-1");
    }
}

interface Interface2 {
    public default void sayHi() {
        System.out.println("Hi Interface-2");
    }
}
public class MyClass implements Interface1, Interface2 {
    public static void main(String[] args) {
        Interface1 obj = new MyClass();
        obj.sayHi();
    }
    public void sayHi() {
        System.out.println("Hi MyClass");
    }
}
```

What is the result?

○ A) Hi Interface-2

○ B) Hi Interface-1

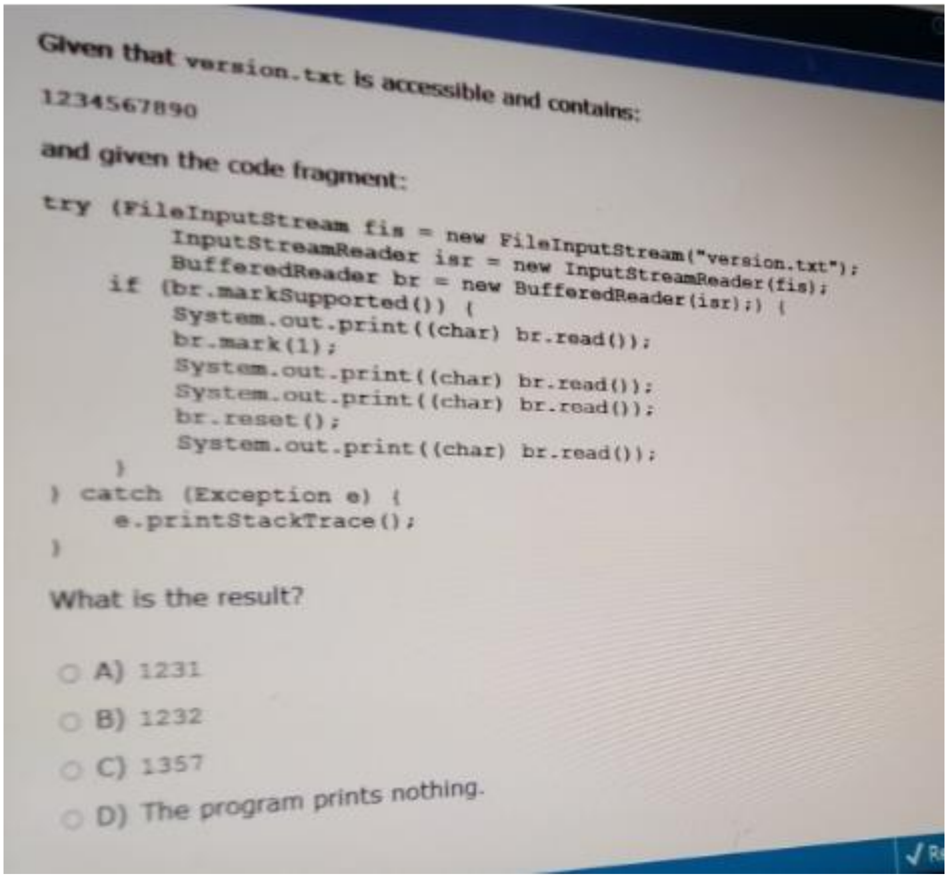
● C) Hi MyClass

○ D) A compilation error occurs.

Correct Answer: C

```
package com.coderbd.Q81;

interface Interface1{

    public default void sayHi(){

        System.out.println("Hi Interface-1");
    }
}
interface Interface2{
    public default void sayHi(){
        System.out.println("Hi Interface-2");
    }
}
public class MyClass implements Interface1, Interface2{
    public static void main(String[] args){
        Interface1 obj= new MyClass();
        obj.sayHi();
    }
    public void sayHi(){
        System.out.println("Hi MyClass");
    }

    // Ans: C

}
```

Problem 82:



Given that version.txt is accessible and contains:

1234567890

and given the code fragment:

```
try (FileInputStream fis = new FileInputStream("version.txt");
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);) {
    if (br.markSupported()) {
        System.out.print((char) br.read());
        br.mark(1);
        System.out.print((char) br.read());
        System.out.print((char) br.read());
        br.reset();
        System.out.print((char) br.read());
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

What is the result?

○ A) 1231

○ B) 1232

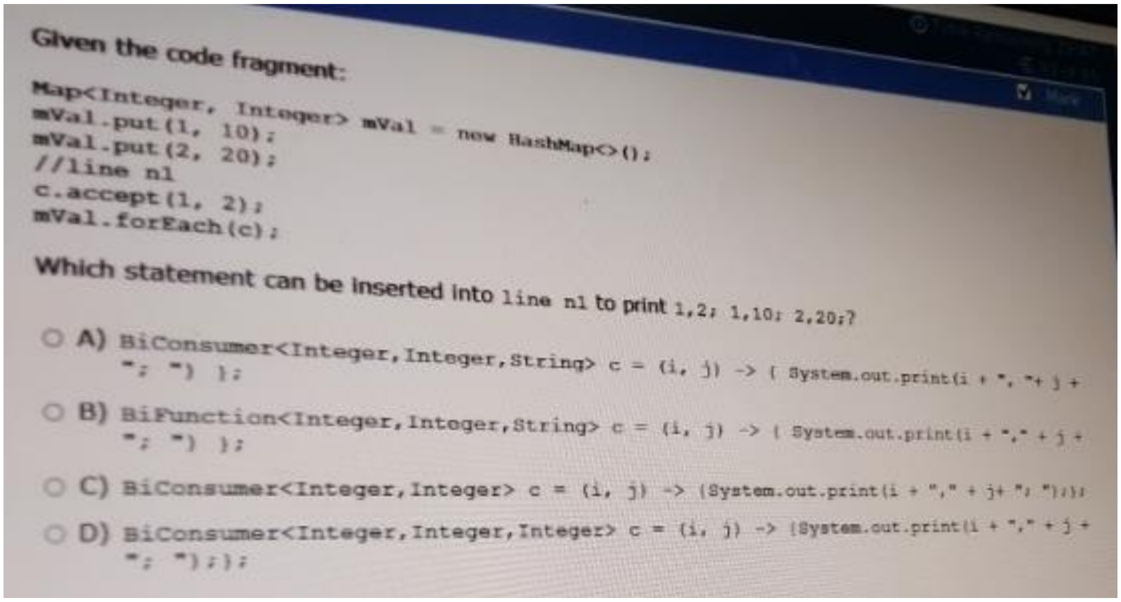
○ C) 1357

○ D) The program prints nothing.

Correct Answer: B

```
package com.coderbd.Q82;

// 1234567890


import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStreamReader;


public class Test {

  public static void main(String[] args) throws IOException {

    try (FileInputStream fis=new FileInputStream("version.txt");

        InputStreamReader isr=new InputStreamReader(fis);)

        BufferedReader br=new BufferedReader(isr);{
        if(br.markSupported()){
           System.out.println((char) br.read());
           br.mark(1);
           System.out.println((char) br.read());
           System.out.println((char) br.read());
           br.reset();
        }

    }catch(Exception e){
        e.printStackTrace();
        }
  }
}  //Ans: Confuse
```

Problem 83:



Given the code fragment:

```
Map<Integer, Integer> mVal = new HashMap<>();
mVal.put(1, 10);
mVal.put(2, 20);
//line n1
c.accept(1, 2);
mVal.forEach(c);
```

Which statement can be inserted into `line n1` to print `1,2; 1,10; 2,20;`?

○ A) `BiConsumer<Integer,Integer,String> c = (i, j) -> { System.out.print(i + ", "+ j + "; ") };`

○ B) `BiFunction<Integer,Integer,String> c = (i, j) -> { System.out.print(i + "," + j + "; ") };`

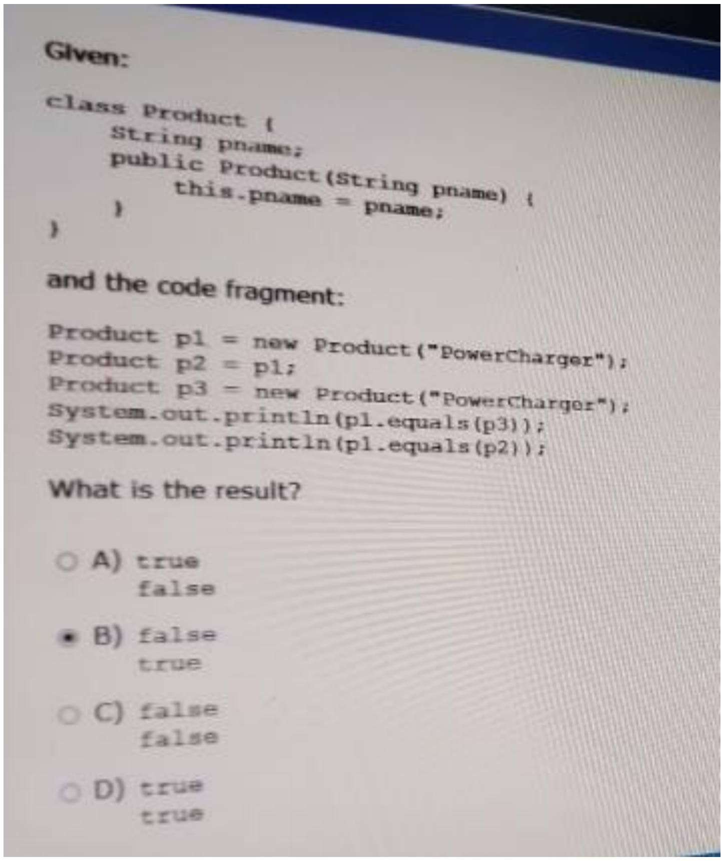
○ C) `BiConsumer<Integer,Integer> c = (i, j) -> {System.out.print(i + "," + j+ "; ");};`

○ D) `BiConsumer<Integer,Integer,Integer> c = (i, j) -> {System.out.print(i + "," + j + "; ");};`

Correct Answer: C

```
package com.coderbd.Q83;

import java.util.HashMap;
import java.util.Map;
import java.util.function.BiConsumer;
import java.util.function.BiFunction;

/**
 *
 * @author Touhid
 */
public class Test {

    public static void main(String[] args) {
        Map<Integer, Integer> mVal = new HashMap<>();
        mVal.put(1, 10);
        mVal.put(2, 20);
        //line n1
//        BiConsumer<Integer,Integer,String> c=(i,j)-> {
//            System.out.println(i+", "+j+";");
//        };
//       BiFunction<Integer,Integer,String> c = (i,j)->{
//           System.out.println(i+", "+j+"; ")
//       };
BiConsumer<Integer,Integer> c = (i,j)->{
           System.out.println(i+", "+j+"; ");
       };
//BiConsumer<Integer,Integer,Integer> c = (i,j)->{
//           System.out.println(i+", "+j+"; ");
//       };
        c.accept(1, 2);
        mVal.forEach(c);


    }
    //Ans: C
}
```

Problem 84:



Given:

```
class Product {
    String pname;
    public Product(String pname) {
        this.pname = pname;
    }
}
```

and the code fragment:

```
Product p1 = new Product("PowerCharger");
Product p2 = p1;
Product p3 = new Product("PowerCharger");
System.out.println(p1.equals(p3));
System.out.println(p1.equals(p2));
```

What is the result?

○ A) true
false

● B) false
true

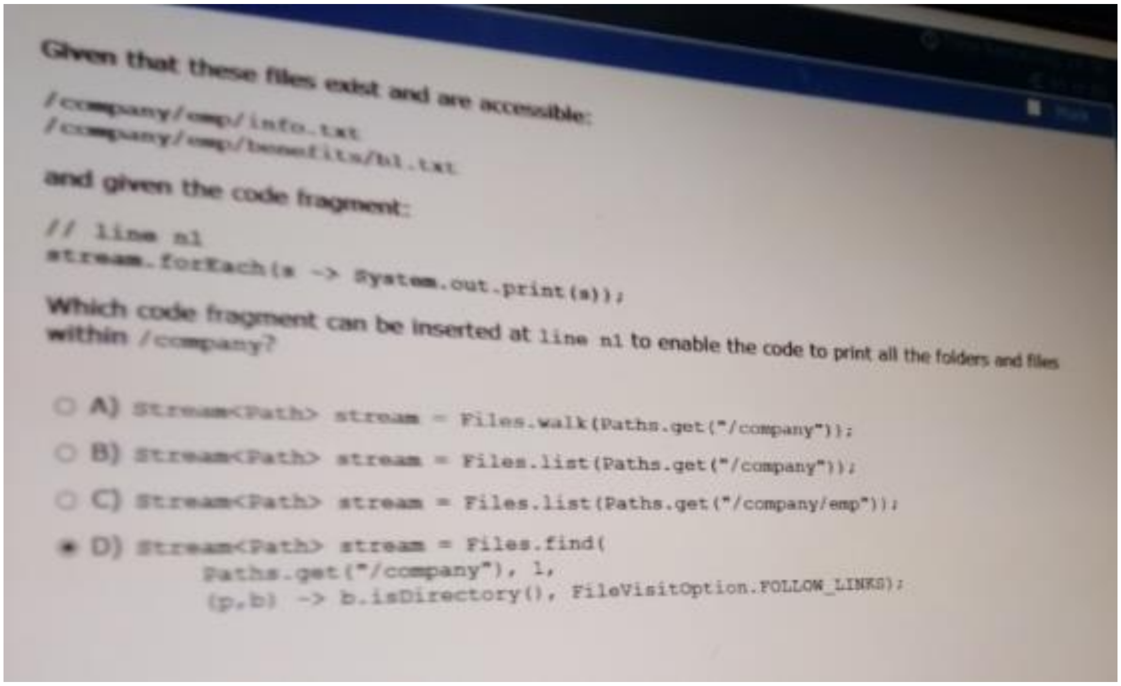
○ C) false
false

○ D) true
true

Correct Answer: B

```
package com.coderbd.Q84;


public class Product {
  String pname;
  public Product(String pname){
    this.pname=pname;
  }
  public static void main(String[] args) {
    Product p1=new Product("PowerCharacter");
    Product p2=p1;
    Product p3=new Product("PowerCharger");
    System.out.println(p1.equals(p3));
    System.out.println(p2.equals(p2));
  }
  // Ans: B
}
```

Problem 85:



Given that these files exist and are accessible:

```
/company/emp/info.txt
/company/emp/benefits/bl.txt
```

and given the code fragment:

```
// line n1
stream.forEach(s -> System.out.print(s));
```

Which code fragment can be inserted at line n1 to enable the code to print all the folders and files within /company?

○ A) `Stream<Path> stream = Files.walk(Paths.get("/company"));`

○ B) `Stream<Path> stream = Files.list(Paths.get("/company"));`

○ C) `Stream<Path> stream = Files.list(Paths.get("/company/emp"));`

● D)
```
Stream<Path> stream = Files.find(
        Paths.get("/company"), 1,
        (p,b) -> b.isDirectory(), FileVisitOption.FOLLOW_LINKS);
```

Correct Answer: A

```
package com.coderbd.Q85;
///     /company/emp/info.txt
///     /company/emp/benefits/bl.txt

import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.stream.Stream;

public class Test {
    public static void main(String[] args) {
        Stream<Path> Stream =Files.walk(Paths.get("/company"));
    }
    // Ans: confuse
}
```