Started on	Friday, 29 January 2021, 09:00
State	Finished
Completed on	Friday, 29 January 2021, 13:13
Time taken	4 hours 13 mins
Marks	181.00/279.00
Grade	**64.87** out of 100.00

Question 1

Partially correct

Mark 60.00 out of 100.00

You will receive points for this problem after the exam is over. Each solved subtask awards some of the points, even if you do not complete all subtasks.

In this problem, you will write the classes for modelling an inventory management system and cash register for a small grocery store. You are allowed, and in some cases required, to add fields and helper methods beyond what is mentioned by each subtask. Each completed subtask below will award points, so you do not need to solve all tasks. Each task corresponds to a test, and you may submit your solution as many times as you like without point deductions.

All fields must be private and all methods and constructors must be public. The return type of methods is void, if no return value is mentioned.

Subtasks:

1. Implement the **SaleItem** class to represent an item in stock at the store. A SaleItem has a name (a text string), a category (another String), a priceInCents (a whole number), and a stock (another whole number) indicating how many items are currently for sale in the store.
2. Implement the constructor to **SaleItem** such that it initializes the fields, with one parameter for each field in the order mentioned above.
3. Add a method **setPrice(price)** that updates the field **price** to the value given as parameter.
4. Implement the method **checkStock(count)** in **SaleItem** such that it returns **true** if there is at least **count** items left in stock, and **false** otherwise.
5. Implement the method **addToStock(extra)** that increases the stock by the extra amount.

The **ShoppingBasket** class should keep track of the items as single customer would like to buy. You are free to choose what fields the class has.

6. Add a method **addItem(item)** that takes a SaleItem and adds a single one of it to the shopping basket. If the customer wants to buy more than one, the same SaleItem object will be added multiple times. Then add a method **totalCost()** that returns the sum of all items in the shopping basket.
7. Implement the method **checkStock()** in **ShoppingBasket** such that it returns **true** or **false** depending on whether there are enough items in stock to purchase all the items in the shopping basket. The method must not change the current stock of any item.

Two exceptions, **InvalidPurchase** and **OutOfStock** have been defined in the code template.

8. Add a method **finalizePurchase()** that locks the order, and removes the items from stock (i.e., have their stock count reduced). If any item has too few items in stock, an **OutOfStock** exception must be thrown (you choose the message), and no items must be removed.
9. Modify **finalizePurchase** so that if it is called multiple times, only the first call must do something; you can only finalize once.
10. Modify **addItem(item)** so that if the purchase has already been finalized, it throws an InvalidPurchase exception (you choose the message) and does not add the item.

Invalid archase exception (you choose the message) and does not add the item.

11. Create a method **printReceipt()** that calls **finalizePurchase** (propagating any exception) and prints a list of all the items in the basket, sorted primarily by category, secondly by name. Each line must contain the name of the item followed by the price in dollars. Finally, a total must be printed.

Example:

```
Bread $1.29
Cheese $4.00
Milk $0.99
Milk $0.99
---
Total: $7.27
```

Answer: (penalty regime: 0 %)

<button>Reset answer</button>

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
class SaleItem {

    private String name;
    private String category;
    private int priceInCents;
    private int stock;

    public SaleItem(String name, String category, int priceInCents, int
stock){
        this.name = name;
        this.category = category;
        this.priceInCents = priceInCents;
        this.stock = stock;
    }

    public void setPrice(int price){
        priceInCents = price;
    }
}
```

	Test	Expected	Got	
✕	subTask("all");	Subtask 1 complete! Subtask 2 complete! Subtask 3 complete! Subtask 4 complete! Subtask 5 complete! Subtask 6 complete! Subtask 7 complete! Subtask 8 complete! Subtask 9 complete!	Subtask 1 complete! Subtask 2 complete! Subtask 3 complete! Subtask 4 complete! Subtask 5 complete! Subtask 6 complete! Subtask 7 incomplete: Reflection invocation failed Subtask 8 incomplete: InvalidPurchase does not have a constructor with the right signature. Check the problems description again. Subtask 9 incomplete: finalizePurchase should not throw an exception on a valid purchase Subtask 10 incomplete: finalizePurchase should not throw an exception on a valid purchase	✕

		complete!	purchase	
		Subtask 10 complete! Subtask 11 complete!	Subtask 11 incomplete: ShoppingBasket does not have a "printReceipt" method with the right signature. Check the problem description again.	

Show differences

Question author's solution (Java):

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```java
import java.util.*;

class SaleItem {
  private String name;
  private String category;
  private int priceInCents;
  private int stock;

  public SaleItem(String name, String category, int priceInCents, int stock) {
    this.name = name;
    this.category = category;
    this.priceInCents = priceInCents;
    this.stock = stock;
  }

  public String getName() {
    return name;
```

Partially correct

Marks for this submission: 0.00/100.00.

Comment:

Question **2**

Correct

Mark 20.00 out of 20.00

When filling out a form with user names, students have used different syntax. Some have used upper case letters, and some have included "**@itu.dk**" after the user name. You would like to compute the corresponding list of email addresses, each of which should consist of a lower case user name followed by "**@itu.dk**".

Write a method **String usernameToEmail(String username)** that converts a student user name in any of the mentioned formats into the corresponding email address.

For example:

Test	Result
System.out.println(usernameToEmail("keUw")); System.out.println(usernameToEmail("kviwro")); System.out.println(usernameToEmail("bkapt@itu.dk"));	keuw@itu.dk kviwro@itu.dk bkapt@itu.dk

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
String usernameToEmail(String username){
      String result = "";
      // String lastChars = username.substring(username.length() - 7);

      username = username.toLowerCase();




      if (username.endsWith("@itu.dk")) {
         result = username;
      } else result = username + "@itu.dk";

      return result;


   }
```

	Test	Expected	Got
✓	System.out.println(usernameToEmail("keUw")); System.out.println(usernameToEmail("kviwro")); System.out.println(usernameToEmail("bkapt@itu.dk"));	keuw@itu.dk kviwro@itu.dk bkapt@itu.dk	keuw@itu. kviwro@it bkapt@itu

Passed all tests! ✓

Question author's solution (Java):

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
public static String usernameToEmail(String username) {
    if (username.contains("@itu.dk")) {
        return username.toLowerCase();
    }
    else {
        return username.toLowerCase()+"@itu.dk";
    }
}
```

Correct

Marks for this submission: 20.00/20.00.

Question 3

Correct

Mark 20.00 out of 20.00

Suppose you have a list of ITU user names collected from a questionnaire:

```
ArrayList<String> usernames;
```

Students have used different syntax: Some have used upper case letters, and some have included "@itu.dk" after the user name. You would like to compute the corresponding list of email addresses, each of which should consist of a lower case user name followed by "@itu.dk". If you have already written a method String usernameToEmail(String username) that converts a

student user name in any of the mentioned formats into the corresponding email address, you can reuse it in this problem.

Write a method with signature
`List<String> usernameListToEmails(List<String> usernames)`
that takes an arbitrary list of user names and creates a list of the corresponding email addresses. The input list **usernames** should not be changed.

The hidden test method **usernameListToEmailsTest** prints the generated email addresses and compares to the list computed by your method. The first test cases are visible, but other test cases are hidden.

For example:

Test	Result
var usernames = new ArrayList<String>(Arrays.asList("keUw", "kviwro", "bkapt@itu.dk")); usernameListToEmailsTest(usernames);	keuw@itu.dk kviwro@itu.dk bkapt@itu.dk

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
List<String> usernameListToEmails(List<String> usernames) {
    List<String> emails = new ArrayList<>();

    for (String username : usernames) {
        String result = "";
        username = username.toLowerCase();

        if (username.endsWith("@itu.dk")) {
            result = username;
        } else result = username + "@itu.dk";
        emails.add(result);

    }

    return emails;

}
```

	Test	Expected	Got	
✓	var usernames = new ArrayList<String> (Arrays.asList("keUw", "kviwro", "bkapt@itu.dk")); usernameListToEmailsTest(usernames);	keuw@itu.dk kviwro@itu.dk bkapt@itu.dk	keuw@itu.dk kviwro@itu.dk bkapt@itu.dk	✓

Passed all tests! ✓

Question author's solution (Java):

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
public static String usernameToEmail(String username) {
    if (username.contains("@itu.dk")) {
        return username.toLowerCase();
    }
    else {
        return username.toLowerCase()+"@itu.dk";
    }
}

public static List<String> usernameListToEmails(List<String> usernames) {
    List<String> result = new ArrayList<String>();
    for (var s: usernames) {
        result.add(usernameToEmail(s));
    }
    return result;
}
```

Correct

Marks for this submission: 20.00/20.00.

Question 4

Partially correct

Mark 10.00 out of 20.00

You must write a method called **printTriangle** that takes a single positive integer n as a parameter. It must then print out the following pattern:

```
1
2 1
...
n−2 ... 2 1
n−1 n−2 ... 2 1
n n−1 n−2 ... 2 1
n−1 n−2 ... 2 1
n−2 ... 2 1
...
2 1
1
```

Note that there has to be a space before the first number in each line. If you wish, it is fine to write other methods that are called by **printTriangle**. Just place all methods next to each other.

For example:

Test	Result
printTriangle(4);	1 2 1 3 2 1 4 3 2 1 3 2 1 2 1 1

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
public void printTriangle(int n){
        int numRows = n + 2;

    System.out.println(" " + (n-3));
```

```
            System.out.println(" " + (n- 2) + " " + (n-3));
            System.out.println(" " + (n-1) + " " + (n- 2) + " " + (n-3));
            System.out.println(" " + n + " " + (n-1) + " " + (n- 2) + " "
+ (n-3));
            System.out.println(" " + (n-1) + " " + (n- 2) + " " + (n-3));
            System.out.println(" " + (n- 2) + " " + (n-3));
            System.out.println(" " + (n-3));


    }
```

	Test	Expected	Got	
✓	`printTriangle(4);`	1 2 1 3 2 1 4 3 2 1 3 2 1 2 1 1	1 2 1 3 2 1 4 3 2 1 3 2 1 2 1 1	✓
✗	`printTriangle(1);`	1	−2 −1 −2 0 −1 −2 1 0 −1 −2 0 −1 −2 −1 −2 −2	✗
✗	`printTriangle(11);`	1 2 1 3 2 1 4 3 2 1 5 4 3 2 1 6 5 4 3 2 1 7 6 5 4 3 2 1 8 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 10 9 8 7 6 5 4 3 2 1 11 10 9 8 7 6 5 4 3 2 1 10 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 8 7 6 5 4 3 2 1 7 6 5 4 3 2 1 6 5 4 3 2 1 5 4 3 2 1 4 3 2 1 3 2 1 2 1 1	8 9 8 10 9 8 11 10 9 8 10 9 8 9 8 8	✗

Some hidden test cases failed, too.
Your code must pass all tests to earn any marks. Try again.

Show differences

Question author's solution (Java):

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
void printDecreasing(int n) {
    for (int i=n; i>0; i--) System.out.print(" "+i);
    System.out.println();
}


void printTriangle(int n) {
    for (int i=1; i<n; i++) printDecreasing(i);
    for (int i=n; i>0; i--) printDecreasing(i);
}
```

Partially correct

Marks for this submission: 0.00/20.00.

Comment:

Question 5

Correct

Mark 4.00 out of 4.00

Assume the definition for **A** starts with:
public abstract class A implements B {...
Which of the following are consequences of introducing the code **new A()** inside the **main** method?

Select one or more:

☑ a.   A compile error occurs as **A** is an abstract class   ✓

☐ b.   An instance of **A** is created

☐ c.   Instances of **A** as well as **B** are created

☐ d.   A compile error occurs as the constructor has no parameters

The correct answer is: A compile error occurs as **A** is an abstract class

---

Question 6

Correct

Mark 4.00 out of 4.00

Which of the following declares and constructs an **ArrayList** with an initial capacity of 10, which can contain instances of **Object**.

Select one or more:

☐ a.   ArrayList list[10] = new ArrayList(Object);

☐ b.   List Object[10] = new ArrayList();

☑ c.   var list = new ArrayList<>();   ✓

☑ d.   ArrayListObject> list = new ArrayListObject>(10);   ✓

The correct answers are: **ArrayListObject> list = new ArrayListObject>(10);**, **var list = new ArrayList<>();**

Question 7

Incorrect

Consider the code fragment:
String x = "abc";

Mark 0.00 out
of 4.00

```
String[] myArray = {x, x};
x = x+"de";
myArray[0] = myArray[0] + "fg";
```
Which of the following expressions are true after running this code?

Select one or more:

☐ a. myArray[1] == "abcdefg"

☐ b. myArray[0] == myArray[1]

☑ c. myArray[1] == "abcde"   ✗

☐ d. myArray[0] == "abcfg"

The correct answer is: **myArray[0] == "abcfg"**

Question 8

Incorrect

Mark 0.00 out
of 4.00

Which of the following statements about Java **assert** statements are true?

Select one or more:

☑ a. An assertion that is not true throws an error   ✓

☐ b. Assertions should have side effects

☑ c. Assertions should be removed before releasing the final code   ✗

☐ d. The Java runtime machine has a flag related to assertions

The correct answers are: An assertion that is not true throws an error, The Java runtime
machine has a flag related to assertions

Question 9

Incorrect

Mark 0.00 out
of 4.00

Consider the class
```
public class myClass {
public static int a = 42;
public int b = 142;
}
```
and the code snippet
```
x = new myClass(); y = new myClass();
x.a += 1; x.b += 1; y.a += 1; y.b += 1;
```
Which of the following expressions are true after running the code?

Select one or more:

☐ a. x.a == 44

☑ b. y.a == 43   ✗

☐ c. x.b == 144

☑ d. y.b == 143   ✓

The correct answers are: **x.a == 44, y.b == 143**

Question 10

Not answered

Marked out of 4.00

Consider classes
public class A {...}
public class B extends A {...}
public class C extends A {...}
Which of the following snippets results in a *run-time* error?

Select one or more:

- ☐ a.   A xa = new B(); C xc = new C(); B xb = (B)xa;

- ☐ b.   B xb = new B(); C xc = new C(); A xa = (A)xc;

- ☐ c.   B xb = new B(); A xa = (A)xb; C xc = (C)xa;

- ☐ d.   B xb = new B(); A xa = xb; C xc = new C();

The correct answer is: **B xb = new B(); A xa = (A)xb; C xc = (C)xa;**

Question 11

Correct

Mark 4.00 out of 4.00

Consider an **Integer** object **n** containing a negative number. Which of the following can happen (for some negative value of **n**) when executing the code **n = n - 1**?

Select one or more:

- ☐ a.   n becomes zero

- ☐ b.   n becomes NaN

- ☐ c.   An ArithmeticException is raised

- ☑ d.   n becomes a positive number   ✓

The correct answer is: **n** becomes a positive number

Question 12

Correct

Mark 4.00 out of 4.00

Consider a **Double** object **x** containing a positive number. Which of the following can happen (for some positive, finite value of **x**) when executing the code **x = 2 * x** ?

Select one or more:

- ☐ a.   x becomes a negative number

- ☐ b.   An ArithmeticException is raised

- ☐ c.   x becomes equal to Double.NaN

- ☑ d.   x becomes equal to Double.POSITIVE_INFINITY   ✓

The correct answer is: **x** becomes equal to **Double.POSITIVE_INFINITY**

Question 13

Correct

Mark 4.00 out of 4.00

Which of the following Java keywords can be used to achieve *looser coupling* between classes?

Select one or more:

- ☐ a.   static

- ☐ b.   public

☐ c. final

☑ d. private ✓

The correct answer is: **private**

Question 14

Partially correct

Mark 2.67 out of 4.00

For **Integer** objects **i** and **j**, which of the following types of exception can be thrown when evaluating the expression **i/(j+1)**?

Select one or more:

☑ a. IllegalArgumentException ✗

☐ b. IOException

☐ c. IndexOutOfBoundsException

☑ d. ArithmeticException ✓

The correct answer is: **ArithmeticException**

Question 15

Partially correct

Mark 2.67 out of 4.00

Assume that in class **A** we have a field such as **final static int[] nums = { 2, 3, 5}**. Assume method **a** is implemented to change value **5** to **7**. Which of the following options are correct?

Select one or more:

☑ a. The method can use **A.nums[2]=7;** ✓

☐ b. This is not possible since nums is final

☐ c. This is not possible since nums is static

☑ d. The method can use nums = { 2, 3, 7} to apply the change ✗

The correct answer is: The method can use **A.nums[2]=7;**

Question 16

Partially correct

Mark 2.00 out of 4.00

Which of the following variables can be set to the value **null**?

Select one or more:

☐ a. String y;

☐ b. boolean w;

☑ c. Double z; ✓

☐ d. int x;

The correct answers are: **String y;**, **Double z;**

Question 17

A class **A** and its subclass **B** both have a method **a()** implemented. If **b** refers to an object of type **B**,

Correct

Mark 2.00 out of 2.00

what will the code **b.a()** do?

Select one or more:

☑ a.   The **a()** method defined in **B** will be called   ✓

☐ b.   The **a()** method defined in **A** will be called

☐ c.   A compiler error will occur as **a()** has been defined twice

☐ d.   Overloading will be used to decide which **a()** is called

---

The correct answer is: The **a()** method defined in **B** will be called

---

Question **18**

Correct

Mark 5.00 out of 5.00

Consider classes:
public abstract class C {...} and
private final class D extends C {...}.
Which of the following statements are true?

Select one or more:

☐ a.   It is possible to create an instance of class **C**.

☐ b.   It is possible to create a subclass **F** of class **D**.

☑ c.   It is possible to create another subclass **E** of class **C**.   ✓

☐ d.   It is possible to create additionally another superclass **G** for class **D**.

☑ e.   It is possible to create an instance of class **D**.   ✓

The correct answers are: It is possible to create an instance of class **D**., It is possible to create another subclass **E** of class **C**.

Question **19**

Correct

Mark 4.00 out of 4.00

Consider the following class definition:

```
public class Mysterious{
  public int x = 3;
  private int y = 1;
  public void setX(int z){
    x += z;
  }
  public void setY(int z){
    y = z;
  }
}
```

What should be changed for the above to enforce encapsulation?

Select one or more:

☐ a.   change both **setX** and **setY** to **private**

☐ b.   change only **setY** to **private**

☐ c.   change **Mysterious** to **private**

☑ d.   change **x** to **private**   ✓

The correct answer is: change **x** to **private**

Question **20**

Correct

Mark 2.00 out of 2.00

Which of the following class headers correctly defines a subclass of **H** that supports the interface **I**?

Select one or more:

☐ a. public class myClass extends I throws H

☐ b. public class myClass extends H throws I

☑ c. public class myClass extends H implements I ✓

☐ d. public class myClass extends I implements H

The correct answer is: **public class myClass extends H implements I**

Question **21**

Correct

Mark 4.00 out of 4.00

Suppose **x** supports the **Collection** interface, and that **x.get(2)** returns **3** at some point in the code. Which of the following are a possible types for **x**?

Select one or more:

☐ a. SetInteger,Integer>

☑ b. MapInteger,Integer> ✓

☑ c. ListInteger> ✓

☐ d. ListListInteger>>

The correct answers are: ListInteger>, MapInteger,Integer>

Question **22**

Correct

Mark 4.00 out of 4.00

Consider the following code fragment
Car c = new Car("red", 2006);
c.register();
Which of the following statements are true about this code?

Select one or more:

☑ a. The constructor method has two parameters ✓

☐ b. **c** refers to an object, **register** is a constructor, **Car** is a class

☐ c. **Car** is a constructor, **c** is a method, **register** is an object

☑ d. **Car** is a class, **c** refers to an object, **register** is a method ✓

The correct answers are: **Car** is a class, **c** refers to an object, **register** is a method, The constructor method has two parameters

---

Question **23**

Incorrect

Which of the following statements are correct?

Mark 0.00 out
of 4.00

Select one or more:

☑ a.   Running:
        try{}catch(E1 e2) {...} catch(E2 e2) {...} finally {System.out.println("finally")},   ✓
        always finishes with printing *finally*

☑ b.   Any statement that can throw an Exception must be enclosed in a try block   ✗

☐ c.   catch(E e) can catch subclasses of E and E is a subclass of Throwable

☐ d.   If Exception e=InvalidStateException("some message") then the method

        int my Method(int x) {

         throw e

        return 0

        }

        The return type is InvalidStateException exception

The correct answers are: Running:
try{}catch(E1 e2) {...} catch(E2 e2) {...} finally {System.out.println("finally")},
always finishes with printing *finally*, catch(E e) can catch subclasses of E and E is a subclass of
Throwable

Question **24**

Incorrect

Mark 0.00 out
of 2.00

Consider the following code fragment:

class A {
int m(ArrayList list) {...return 1;}
}
class B extends A{
int m(List list) {... return 2;}
}
Which of the following options are correct?

Select one or more:

☐ a.   For b (an object of B), a call to **m** can return either **2 or 1.**

☐ b.   For b (an object of B), a call to **m** can only **return 1.**

☑ c.   For b (an object of B), a call to **m** can only return 2.   ✗

☐ d.   For a (an object of A), a call to **m** can return either **2 or 1.**

The correct answer is: For b (an object of B), a call to **m** can return either **2 or 1.**

Question **25**

Correct

Mark 4.00 out
of 4.00

Java **String** objects have the method **public String substring(int beginIndex, int endIndex)** which
returns a **String** object containing a substring with specified start and end positions (not including
the character at **endIndex**). Suppose **String x = "abcdefgh"**, which of the following expressions
returns the string "efg"?

Select one or more:

☐ a.   x.substring(0,7).substring(4,6)

☐ b.   x.substring(3,6).substring(1,7)

☑ c.   x.substring(1,7).substring(3,6)   ✓

☐ d.   x.substring(3,6).substring(0,7)

The correct answer is: **x.substring(1,7).substring(3,6)**

Question 26

Incorrect

Mark 0.00 out of 4.00

Which of the following show correct matching of the following regular expression?

\(.*?\)

Select one or more:

☐ a. $**)(* * * * * *) * *)$

☑ b. $((*) * *))$ ✗

☑ c. $(**)*)$ ✓

☑ d. $(**) * (*))$ ✗

The correct answers are: $(**)*), **)(* * * * * *) * *)$

Question 27

Correct

Mark 4.00 out of 4.00

Which of the following are matched by the regular expression on the next line?

ab*c+

Select one or more:

☐ a.   bbbccc

☑ b.   accc   ✓

☐ c.   aaaabb

☑ d.   abbbc   ✓

The correct answers are: accc, abbbc

Question 28

Correct

Mark 4.00 out of 4.00

Suppose you have a database of 25,033 web pages containing 103,252 distinct words. How many queries to the inverted index are needed to retrieve the two web pages containing the words "**Copenhagen capital Denmark**"?

Select one or more:

☐ a.   2 queries, one per matching web page

☑ b.   3 queries, one per word in the query   ✓

☐ c.   25,033 queries, one per web page in the database

☐ d.   103,252 queries, one per distinct word

The correct answer is: 3 queries, one per word in the query

Question 29

Correct

Mark 2.00 out of 2.00

Which of the following are the purpose of a ranking algorithm?

Select one or more:

☐ a. Allowing the use of the keyword 'OR' in queries

☑ b. Choosing the order in which search results are presented ✓

☐ c. Generating a result in JSON format that can be displayed in the browser

☐ d. Sorting the list of web pages so they can be searched faster

The correct answer is: Choosing the order in which search results are presented

Question 30

Not answered

Marked out of 4.00

Consider the following code. Which one of the following can be the value of **sum** after running:
int sum=0;
int cnt=0;

HashSet<Integer> hashSet=new HashSet<Integer>();

for(int i=1;i<6;i++){

hashSet.add(i*2);

}

for (int t:hashSet){ if(cnt<3){

sum+=t;
cnt++;
}
else{break;}
}

Select one or more:

☐ a. 20

☐ b. 16

☐ c. 15

☐ d. 26

The correct answers are: 16, 20

Question 31

Not answered

Marked out of 4.00

Consider the following code fragment: var userNames = employeeRecords.stream()
?
.collect(toList());
Which of the following could replace **?**, to get the usernames for employees who have salaries greater than or equal to a **Max** value?

Select one or more:

☐ a. .filter(r -> r.getSalary() >= Max) .map(r -> r.getUserName())

☐ b. .map(r -> r.getSalary() >= Max) .filter(r -> r.getUserName())

☐ c. .map(r -> r.getUserName()) .filter(r -> r.getSalary() >= Max)

☐ d.   .filter(r -> r.getUserName()) .map(r -> r.getSalary() >= Max)

The correct answer is: **.filter(r -> r.getSalary() >= Max) .map(r -> r.getUserName())**

Question **32**

Incorrect

Mark 0.00 out of 4.00

Consider the method
public Boolean myMethod(Boolean a, Boolean b) {
if(a==b) return true; else return a;
}
For testing **myMethod** with two test cases, which of the following asserts are correct usage of the API?

Select one or more:

☑ a.   assertEquals(false,myMethod(false,true))   ✗

☑ b.   assertTrue(myMethod(false,true))   ✓

☐ c.   assertFalse(myMethod(false,false))

☐ d.   assertEquals(myMethod(true,false),true)

The correct answers are: **assertTrue(myMethod(false,true)), assertFalse(myMethod(false,false))**

Question **33**

Not answered

Marked out of 4.00

Consider the method
public Boolean myMethod(Boolean a, Boolean b) {
if(a==b) return true; else return a;
}
For testing **myMethod** which of the following assertions will get pass as result?

Select one or more:

☐ a.   assertFalse(myMethod(false,false))

☐ b.   assertTrue(myMethod(false,true))

☐ c.   assertEquals(false,myMethod(false,true))

☐ d.   assertEquals(myMethod(true,false),true)

The correct answers are: **assertEquals(myMethod(true,false),true), assertEquals(false,myMethod(false,true))**

Question **34**

Partially correct

Mark 2.67 out of 4.00

Suppose that **map** has type **HashMap<String, Integer>**. Which of the below statements describes this code?

```
for (var method : map.keySet()) {
    // ...
}
```

Select one or more:

☑ a.   A method that maps a key set   ✗

b.   A loop that iterates over a set of integers

☑ c.   A loop that iterates over a set of strings   ✓

d.   A method that takes a map as an argument

Your answer is partially correct.

You have selected too many options.
The correct answer is: A loop that iterates over a set of strings

Question **35**

Correct

Mark 2.00 out of 2.00

Suppose **myMap** is of type **HashMapInteger, ListString**>>. What is the type of variable **s** in the loop **for (var s : myMap.keySet()) {…}**?

Select one or more:

a.   ListString>

☑ b.   Integer   ✓

c.   int

d.   String

The correct answer is: **Integer**

Question **36**

Correct

Mark 4.00 out of 4.00

Which of the following statements are correct?

Select one or more:

☑ a.   The execution of Java applications begins at method main   ✓

b.   A class which is implementing an interface can implement some of the methods of the interface

c.   A private member of a class cannot be accessed by the methods of the same class

☑ d.   A class containing abstract methods must be an abstract class   ✓

Your answer is correct.

The correct answers are: A class containing abstract methods must be an abstract class, The execution of Java applications begins at method main