CHALMERS

7988870

FÖRSÄTTSBLAD TENTAMEN / EXAMINATION COVER

Kurskod / Course code:	Provkod / Test code:	Tentamensdatum / Examination date:
DIT636	1010	2025-08-27
Anonymt kodnummer / Anonymous code number:	0014-DDU	
Kursnamn / Course name:	Mjukvarukvalitet och testning	

Ifylles av student / To be completed by the student

Behandlade uppgifter. Sätt kryss (X) / Solved assignment. Put an X.:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Antal inlämnade svarsblad. Sätt kryss (X) i rutorna / Number of submitted answer sheets. Put a/an X in boxes.

0	10	20	30	40	50	60	70	80	90		1	2	3	4	5	6	7	8	9
☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	+	☐	☐	☐	☐	☐	☐	☐	☐	☒

Ifylles av lärare / To be completed by the examiner

Poäng på uppgifter / Points per question										
1	2	3	4	5	6	7	8	9	10	Bonus
11	12	13	14	15	16	17	18	19	20	Total
										,

Heltal / integer *0,5*

Datainläsning

Totalpoäng / Total points										
10	20	30	40	50	60	70	80	90	100	200
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
+	0,5	1	2	3	4	5	6	7	8	9
	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Betygskala/ Grade scale	Betyg / Grade			
	U	3	4	5
TH	☐	☐	☐	☐
	U	G	VG	
UV/UG	☐	☐	☐	

7988870

Lärarens kommentarer

1) a MC/DC -2 c

2) a true

3) a true

4) b false

5) b Safetr

6) a true -1 b

7) a Availability
It has to always be available for regulating the temperature to prevent accidents.

3

Availability

Description
Streaming service offers 24/7 video streaming without any down-time

System state: System is running normally with many users logged in watching videos simultaneously.

-1

Load is part of system state. What is "normal" load?

System Environment: System has normal load with many users logged in from browsers.

External Stimulus: One server unexpectedly crashes

Required system respons: Backup server takes over.

Response measure: Backupserver takes over within 20ms.

-3

This scenario is way too broad. You cannot ensure it recovers from any possible error for any possible feature.

-5

No reliability scenario

CHALMERS | Anonymous code | Points for question (to be filled in by teacher) | Consecutive page no. Löpande sid nr 3

Anonym kod: DIT 636 0014-DDU | Poäng på uppgiften (fylles av lärare) | Question no. Uppgift nr 3

Exploratory testing is the test that is done from a users perspective. Unlike the other stages it doesnt have a ready made script but rather goes "with the flow." For inspiration it can be guided by tours that could inspire new ways to think and maybe find new areas to test.
While unit testing usually finds faults in logic and systemtesting usually test if the whole system lives up to the requirements, exploratory testing could find the unexpected bugs and the edgecases.

Somewhat vague. What are the unexpected bugs?

-2

CHALMERS | Anonymous code | Points for question (to be filled in by teacher) | Consecutive page no. Löpande sid nr 4

Anonym kod: DIT636 0014-PDU | Poäng på uppgiften (ifylles av lärare) | Question no. Uppgift nr 4

-2

null or malformed values?

choices	value	tag
session: user ID	"" (empty string)	IF empty → ERROR
	"User 1" (valid)	SINGLE
	"User 2" (does not exist)	IF doesn't exist → ERROR
session: video ID	"" (empty string)	IF empty → ERROR
	"123" (valid)	SINGLE
	"1000 000" (doesn't exist)	IF doesn't exist → ERROR
session: timestamp	"" (empty)	IF empty → ERROR
	360 (valid)	SINGLE
	-100 (faulty)	IF faulty → ERROR
audioLanguage	"" (empty)	IF empty → ERROR
	"a valid one" (valid)	SINGLE
	"a nonexisting one"	IF doesn't exist → ERROR
audioSpeakers	"" (empty)	IF empty → ERROR
	2 (valid)	SINGLE
	200 (too many)	IF to many → ERROR
	-2 (faulty)	IF faulty → ERROR
	2,5 (wrong format)	IF wrong format → ERROR
subtitleLanguage	"" (empty)	IF empty → ERROR
	"a valid one" (valid)	SINGLE
	"a nonexisting one"	IF doesn't exist → ERROR

-1

clarify "faulty"

3

1) The FedEx tour - Just like a package can be traced from sender to reciever we track the datas journy from end-to-end.

2) * Create an account Add valid email, address, password, and credit card number. Test by logging in that the information has been saved in the database and is displayed in the users account.

* Log in to my account. Start watching a video. Stop halfway through and log out. Log in again and open the same video. Check if the timestamp has been saved and it displayes the video in the same timestamp as when you last opened it

* Log in to my account. Save some videos in the watch later list. Check the list to see that they got saved and is displayed there. Watch one of the saved videos to the end. Check the list again to see that the list has been updated and the videos are not displayed anymore.

These sequenses fits because they create data, that should be stored in/removed from the database and displayed in the UI, And you can follow it from creation -> saving -> displaying or deleting -> removed from DB -> not displayed

3

```
@Test
public void testChangeOptionsSuccess(){
  assertTrue(changeOptions(123,"swedish",2,"english");
}
```

Where is "123" defined? -1

```
@Test
public void testChangeOptionsFail(){
  assertFalse(changeOptions(123,"a minority language",2,
                            "english");
}
```

```
@Test
public void testChangeOptionException(){
  assertThrows(FaultySessionException.class, ()->{
    changeOption("xxx","swedish",2,"english");
  });
}
```

-2 Will not compile

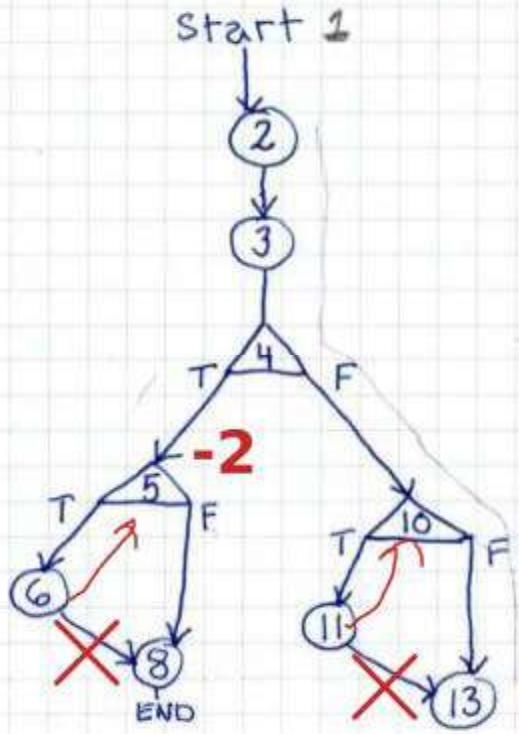
You're passing a string instead of an object

Assumption: FaltySessionException.class exists.
Faulty session creates exception.

CHALMERS | Anonymous code | Points for question (to be filled in by teacher) | Consecutive page no. Löpande sid nr 7

Anonym kod | Poäng på uppgiften (Fylles av lärare) | Question no. Uppgift nr 7

DIT 636 0014-DDU4



T = True

F = False

Test input
1) canPartition([2,3,6,1])
2) canPartition([9,3,4,1])
3) canPartition([10])
4) canPartition([1])

testinputs

lines	1	2	3	4
[4]	4-T	4-F	4-T	4-F
[5]	5-T	–	5-F	–
[10]	–	10-T	–	10-F

3

Def-use pairs

arr: (1,2) (2,4) (2,5) (2,6) (2,8) (2,10) (2,11) (2,13)

product: (3,6) (3,8) (3,11) (3,13) (6,8) (11,13)

(6,6)

(11, 11)

i: (5,6) (10,11)

(5,5), (10, 10)

-3

Test input

1) canPartition([2,3,6,1])
2) canPartition([9,3,4,2])
3) canPartition([0])
4) canPartition([1])

1) arr (1,2) (2,4) (2,5) (2,6) Product: (3,6) (6,8) i (5,6)
2) arr (1,2) (2,4) (2,10) (2,11) product (3,11) (11,13) i (10,11)
3) arr (1,2) (2,4) (2,8) product (3,8)
4) arr (1,2) (2,4) (2,13) product (3,13)

3

Line ↓ example ↓

ROR

~~[3] =~~ **-1**

~~[4] =~~

[4] <

[5] !=

[6] !=

mutant in line [5] change != to ==

[5] if (s1.charAt(index1) == s2.charAt(index2)

test input: oneEditInsert("hej","hhej")

Original output: true

Mutant output: false

AOR

[9] ++

[11] ++

[12] ++

Mutant in line [9] change ++ to --

[9] index2--;

testinput: oneEditInsert("cart","scarf")

Original output: False

Mutant output: True

CRP

[2] 0

[3] 0

[7] false

[15] true

Mutant in line [2] change 0 to 3

[2] int index1 = 3;

test input: oneEditInsert("hi","hello")

original output: False

Mutant output: True

CHALMERS			Question no. Uppgift nr

Question	Your score / Max
Question 1	7 / 10
Question 2	1 / 10
Question 3	6 / 8
Question 4	9 / 12
Question 5	8 / 8
Question 6	6 / 9
Question 7	14 / 16
Question 8	9 / 12
Question 9	14 / 15
Total	**74 / 100**