Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2013-10-23
Sal	TER2
Tid	14-18
Kurskod	TDDD04
Provkod	TEN1
Kursnamn/benämning	Programvarutestning
Institution	IDA
Antal uppgifter som ingår i tentamen	12
Antal sidor på tentamen (inkl. försättsbladet)	7
Jour/Kursansvarig	Kristian Sandahl
Telefon under skrivtid	013-281957
Besöker salen ca kl.	15:00
Kursadministratör (namn + tfnnr + mailadress)	Anna Grabska Eklund
Tillåtna hjälpmedel	Inga

LiTH, Linköpings tekniska högskola
IDA, Institutionen för datavetenskap
David Byers

Written exam

TDDD04 Software Testing

2013-10-23

Permissible aids

Dictionary (printed, NOT electronic)

Teacher on duty

Kristian Sandahl, tel. 013-281957

Instructions and grading

You may answer in Swedish or English.

Your grade will depend on the total points you score on the exam. The maximum number of points is 90. The following grading scale is **preliminary** and the limits may be lowered after grading.

Grade	**3**	**4**	**5**
Points required	42	63	77

Important information: how your answers are assessed

Many questions indicate how your answers will be assessed. This is to provide some guidance on how to answer each question. Regardless of this it is important that you answer each question completely and correctly.

Several questions ask you to define test cases. In some cases you are asked to provide a minimal set of test cases. This means that you can't remove a single test case from the ones you list and still meet the requirements of the question. Points will be deducted if your set of test cases is not minimal. (Note that "minimal" is not the same as "smallest number"; even when it would be possible to satisfy requirements with a single test case, a set of two or three could still be minimal.)

You may find it necessary to make assumptions in order to solve some problems. In fact, your ability to recognize and adequately handle situations where assumptions are necessary (e.g. requirements are incomplete or unclear) will be assessed as part of the exam. If you make assumptions, ensure that you satisfy the following requirements:

- You have documented your assumptions clearly.
- You have explained (briefly) why it was necessary to make the assumption.

Whenever you make an assumption, stay as true to the original problem as possible.

You don't need to be verbose to get full points. A compact answer that hits all the important points is just as good – or better – than one that is long and wordy. Compact answers also happen to be quicker to write (and grade) than long ones.

Please double-check that you answer the entire question. In particular, if you don't give a motivation or example when asked for one, a significant number of points will always be deducted.

Question 1: Terminology (4p)

Explain what "white-box testing" is. Briefly explain one test case design methodology that can be used for white-box testing. (4p)

Question 2: Coverage criteria (8p)

a) If a set of test cases achieves branch coverage does it also achieve condition coverage? (Yes/no is sufficient; 2p)

b) If a set of test cases achieves MCC, does it also achieve MCDC? (Yes/no is sufficient; 2p)

c) Explain ALL-C-USES coverage. It is recommended (but not required) that you illustrate your explanation with a simple example. (4p)

Question 3: Test automation (6p)

Briefly explain the following types of test data (input) generation:

a) Code-based input generation (2p)

b) Interface-based input generation (2p)

c) Specification-based input generation (2p)

Question 4: Easy points (6p)

Answer true or false:

a) The goal of system testing is to verify the interaction between units of the system.

b) Unit testing is usually done before integration testing.

c) A symptom is the observable effect of executing a bug.

d) Code inspections are an effective way to find faults.

e) Domain analysis is a black-box test case design technique.

f) The control script for data-driven test scripts requires little effort to set up.

(It's not worth guessing: you get 1p for correct answer, 0p for no answer, and -1p for incorrect answer; you can get negative points on this question.)

Question 5: Black-box testing (16p)

A control system for a heater operates according to the following rules (where T is the target temperature set by the user):

- If the detected temperature is more than T+5, turn the heater on.
- If the detected temperature is less than T-5, turn the heater off.
- If T is less than 50 or more than 350, indicate an error condition.

Select a test case design methodology for this problem and use it to define a minimal set of test cases (according to that methodology). Motivate your choice of methodology! You will be evaluated on both your choice of methodology, motivation, and application of the methodology.

Question 6: Basis path testing (10p)

Use *basis path testing* to define test cases for the following programs.

```
int f(int x, int y) {
  bool x_neg;
  if (x > 0) {
    x_neg = false;
  }
  else {
    x_neg = true;
    x = -x;
  }

  if (y > 0) {
    if (x_neg) {
      y = -(y * y);
    }
    else {
      y = y * y;
    }
  }
  else {
    y = -(y * y);
  }
  return x + y;
}
```

You are required to draw the appropriate graphical representation of the program and illustrate the process of selecting paths. For each test case, indicate which basis path the test case corresponds to, in addition to the other information required for a test case. You will be assessed on the quality and completeness of your test cases, as well as your explanation of the basis path selection process.

Question 7: State transition testing (10p)

Explain the following types of coverage metrics for state-transition testing:

a) All-states coverage (2p).

b) All-transitions coverage (2p).

Draw a state-transition diagram (any syntax you like, any structure you want) in which each state has a unique label and each transition is taken on a specific input. Using this diagram, create input sequences that achieve:

i. All-states coverage, but not all-transitions or 2-switch (2p).

ii. All-transitions and all-states coverage, but not 2-switch (2p).

iii. All-transitions, all-states and 2-switch coverage (2p).

(*N*-switch coverage means that all paths of length *N* in the state-transition diagram have been taken, so 2-switch coverage means that all paths of length 2 have been taken.)

Question 8: Integration testing (6p)

a) State one advantage big-bang integration has over most other methods. (2p)

b) In the context of integration testing, what is a *driver*. (2p)

c) In the context of integration testing, what is a *stub*. (2p)

Question 9: Exploratory testing (6p)

a) What distinguishes exploratory testing from ad-hoc testing? (2p)

b) In terms of exploratory testing, what is a *charter*? (2p)

c) In terms of exploratory testing, what is a *tour*? (2p)

Question 10: Multiple condition coverage (10p)

Specify a minimal set of test cases for the following function that result in 100% multiple condition coverage.

```
int rules(int a, int b, int c, int d) {
  if (a || b) {
    if (b || c) {
      return 1;
    }
    return 2;
  }
  else if (c && d) {
    return 3;
  return 4;
}
```

Question 11: Model-based testing (4p)

Answer the following questions true or false:

a) The model used in model-based testing can model only part of the behavior of the real system.

b) Model-based testing is suitable for functional testing.

c) Model-based testing is suitable for many large, complex systems.

d) Model-based testing is suitable for many small, fairly simple, systems.

(It's not worth guessing: you get 1p for correct answer, 0p for no answer, and -1p for incorrect answer; you can get negative points on this question.)

Question 12: More easy points (4p)

Answer the following questions true or false:

a) Automated testing is effective for ensuring that old bugs are not reintroduced.

b) Automated tests are repeatable and cheap to run.

c) Automated testing is appropriate in an agile environment.

d) By automated testing people usually mean testing where tests are run automatically, but inputs and outputs are created by hand.

(It's not worth guessing: you get 1p for correct answer, 0p for no answer, and -1p for incorrect answer; you can get negative points on this question.)