THE UNIVERSITY OF HULL

Department of Computer Science

Level 6 Examination

Semester 1

2015/16

Advanced Software Engineering

2 hours

Answer ALL questions
(Each question is out of 20 marks)

You should answer all compulsory questions. If you do not attempt to answer a compulsory question you will receive a mark of 0 for that question.

If you have a choice of questions and you answer more than you are asked to, your answers will be marked in the order that the questions appear on the examination question paper. Any additional questions that you attempt will not be marked.

You should cross out any questions which you attempt but do not wish to be marked.

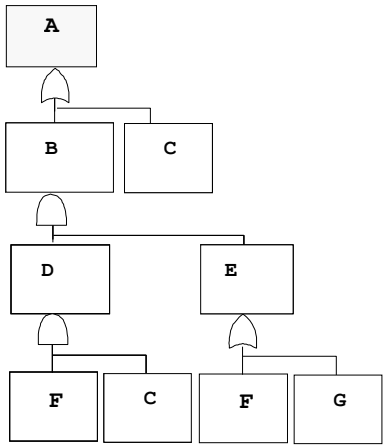
Do not open or turn over this exam paper, or start to write anything until told to by the Invigilator. Starting to write before permitted to do so may be seen as an attempt to use Unfair Means.

1 This question is on techniques for system safety analysis.

(i) Is a reliable system always a safe system? Is an unreliable system always an unsafe system? Justify your answers. [4 marks]

(ii) Give two possible causes for the increasing technological risk in society [2 marks]

(iii) The fault tree below determines how malfunction of component A is ultimately caused by malfunctions of components C, F and G upon which component A relies for correct operation.



(a) Is "C.F" a minimal cut-set of the fault tree and why? Answer this question without calculating all minimal cut-sets. [4 marks]

(b) Calculate all minimal cut-sets. [6 marks]

(c) Among C, F and G, which component is more critical in this design, in the sense that the occurrence of its failure influences more the function of component A? [4 marks]

2. This question is on techniques for software safety analysis and software verification.

(i) Assuming that in the following piece of code

```
while ( j > 6 ) {
   j = j - 1;
   x = x + 10;
}
```

`j` and `x` are integer variables

(a) Draw a software fault tree to show how the unsafe condition "`x>30`" can arise from execution of this code. [6 marks]

(b) If j_0 and x_0 represent the initial values of `j` and `x` before the loop then during the construction of the fault tree you would have to identify combinations of values or relationships between j_0 and x_0 that cause the unsafe output condition "`x>30`". Explain how you have derived those initial conditions that represent basic events in the fault tree structure. [4 marks]

(ii) Given the following program

```
x = 0;
while ( j != 6 ) {
   j = j - 1;
   x = x + 100;
}
```

where `j` and `x` are integer variables

(a) Identify the program input variable or variables. [2 marks]

(b) Give the pre-condition on the input that ensures that the program terminates. [4 marks]

(c) Give the pre-condition on the input that ensures the program post-condition `x < 1001`. [4 marks]

3 This question is on software testing.

(i) Software testing is undecidable.

(a) Explain the above statement. [2 marks]

(b) Explain the practical consequence of undecidability for the construction of software test tools. [3 marks]

(ii) Consider the following program fragment where the input domain for x is {0, 1} and the input domain for y is {-1, 0}.

```
k = x;
if (x == 0) {                                    // if-0
  if (y == 0) {                                  // if-1
    return k;
  }
}
if ((x < 0 && y > 0) || (x > 0 && y < 0)) {  // if-2
  return 3;
}
return 2;
```

(a) Complete the table below to indicate the branch executed at each if-statement in the program above when the test at the head of each column is executed.

	Tests			
	(x = 0, y = -1)	(0, 0)	(1, -1)	(1, 0)
if-0	T			
if-1				
if-2				
returns				

In the bottom row of the table (i.e. "returns") give the value returned for each test. [4 marks]

(b) Give a minimal set of tests that achieve coverage of the feasible branches in the program. [3 marks]

(c) By partitioning the input domain into subsets of inputs that produce the same output, give a minimal set of tests that achieve partition coverage for the program. [4 marks]

(d) Consider a mutation of the program in which the logical operator || is replaced with the logical operator &&. Give a test that will distinguish this mutant from the original program. [4 marks]