



DUBLIN CITY UNIVERSITY

SEMESTER TWO EXAMINATIONS 2011

MODULE:
(Title & Code) CA648 Formal Programming

COURSE: M.Sc. in Software Engineering

YEAR: 1

EXAMINERS:
(Including Telephone Nos.) Dr. A. Butterfield,
Mr. D. Brady,
Dr. G. Hamilton, Ext no. 5017.

TIME ALLOWED: 3 hours

INSTRUCTIONS: Please answer all questions.
All questions carry equal marks.

Please do not turn over this page until instructed to do so

The use of programmable or text storing calculators is expressly forbidden.

QUESTION 1**[TOTAL MARKS: 20]**

Consider the following partial correctness specification:

```
{J = 0 ∧ N ≥ 1}
SUM := 0;
WHILE J < N DO
  BEGIN
    J := J + 1;
    SUM := SUM + (2 × J)
  END
{SUM = N × (N + 1)}
```

1(a) **[6 Marks]**

Add appropriate annotations to this specification to allow it to be verified.

1(b) **[6 Marks]**

List the verification conditions which would be generated for the annotated specification in 1(a).

1(c) **[8 Marks]**

Verify this specification by showing that the verification conditions in 1(b) are true.

QUESTION 2**[TOTAL MARKS: 20]****2(a)** **[4 Marks]**Explain the difference between *partial* and *total* correctness.**2(b)** **[4 Marks]**

Give an example of a specification which is partially correct, but not totally correct.

2(c) **[12 Marks]**

Show that the following annotated specification is true:

```
[X = x ∧ X ≥ 0]
P := 1; {X = x ∧ X ≥ 0 ∧ P = 1}
WHILE X > 0 DO {P = Yx-X ∧ X ≥ 0} [X]
  BEGIN
    X := X - 1;
    P := P × Y
  END
[P = Yx]
```

QUESTION 3**[TOTAL MARKS: 20]****3(a)****[4 Marks]**

Describe how a theory of program refinement can be defined on top of Floyd-Hoare logic.

3(b)**[4 Marks]**

Define the specification notation $[P, Q]$.

3(c)**[12 Marks]**

Refine the following specification to a corresponding program:

$$[X = n \wedge n \geq 0, Y = 2^n]$$
QUESTION 4**[TOTAL MARKS: 20]**

An airline reservation system has a number of flights, each of which has a unique flight number and an associated number of seats. Customers can make reservations for flights, each of which is given a unique booking reference. Customers can also cancel reservations. The following events need to be handled:

reserve: reserve the given number of seats on one flight with the given flight number using the given new booking reference.

cancel: cancel the reservation with the given booking reference.

addflight: add a new flight with the given flight number and number of seats.

removeflight: remove the flight with the given flight number; this event is not enabled if there are any reservations on the flight.

freeseats: return the number of free seats on the flight with the given flight number.

4(a)**[2 Marks]**

Define the context for an Event-B specification of the airline reservation system.

4(b)**[6 Marks]**

Define the variables for an Event-B specification of the airline reservation system. Define a suitable invariant for these variables, and show their initialisation, ensuring that this initialisation satisfies the invariant.

4(c)**[12 Marks]**

Specify the events for an Event-B specification of the airline reservation system, making use of the definitions in 4(a) and 4(b).

QUESTION 5**[TOTAL MARKS: 20]****5(a)****[7 Marks]**

Write an Event-B specification for a program computing the index at which a value v occurs in an array $a : 1..n \rightarrow \mathbb{N}$ where $n \geq 1$ (you can assume that v does occur in a). The specification should define v , a and n as constants and use a result variable $result$. It should also have two abstract events *Initialisation* and *Search* which give the appropriate precondition and postcondition respectively for $result$.

5(b)**[8 Marks]**

Give a refinement of the specification in 5(a) which adds a new variable i , giving the value of the current index in the array. Your refinement should also add one further event *Progress*, a convergent event used to ensure termination by decreasing the variant. You should also refine the events *Initialisation* and *Search* to give precise initial and final values for the $result$ variable.

5(c)**[5 Marks]**

Give a program which computes the index at which a given value occurs in an array and is a refinement of your answers given in 5(a) and 5(b).