BCT 2313: FORMAL SOFTWARE SPECIFICATION METHODS
DUE 20th October 2020 4pm

1. Justify with four areas of commercial use the following statement "Formal specification methods very significant applications in software engineering
(2 Marks)

2. Is the following a tautology? Justify your answer with a truth table

$$[\,(P \Rightarrow Q) \wedge (Q \Rightarrow R)\,] \Rightarrow (P \Rightarrow R)$$ (3 Marks)

3. With aid of a working example, explain the properties of safety and liveness as applied in temporal logic
(2 Marks)

4. Compute the weakest precondition for the following statement and post conditions
a) X=2* y-3 {x>25}
(1 Marks)

b) Compute the weakest precondition fo the following **if** statement:

```
if (a == b)
b = 2 * a + 1;
else
b = 2 * a;
{b > 1}
```
(2 Marks)

5. Describe briefly the following specification paradigms (2 Marks)
   i. History Based Specification
   ii. Transition Based Specifications

6. Using First Order Predicate Calculus, represent the following
   i. All football players are strong
   ii. Nobody likes githeri
   iii. Some people like Omena
   iv. If it does not rain on Monday, Jane will attend Lectures

(2 Marks)

7. Prove by induction that

$$\sum_{r=1}^{n} r^2 = \frac{n}{6}(n+1)(2n+1)$$

for all positive integer values of n. (3 Marks)

8. Write a loop to set sum = 1 + 2 + … + n and prove that it is correct. (3 Marks)

Part 2

1. Describe the Hoares Rules and how they are applied to total and partial specification **10 Marks**
2. Prove the correctness of the following Hoare triple showing the various rules applied . **10 Marks**

```
                { x = x0 }
                    []
                x = x - 3;
               if (x < 0) {
                  x = 1;
                } else {
               if (true) {
                x = x + 1;
                } else {
                 x = 10;
                    }
                    }
{ (x0 < 3 -> x = 1) & (x0 >= 3 -> x < x0) }
```

3. Prove the partial correctness of the following program that divides *a by b, q is* the quotient *and a* is the remainder **5 marks**

```
          {a = _a & b > 0 & a >= 0 }
                    q = 0;
               while (a >= b) {
                  a = a - b;
                  q = q + 1;
                      }
                    }\]
{ q * b + a= _a & a >= 0 & a < b }
```