Institute of Statistical Studies and Research
Department of Computer and Information Sciences
Verification of Computer Programs- **CS601**
January 2008 Time permitted: **Three Hours**

==========

Solve all questions. **Write Complete Answer for each question and Justify it**

==========

Question 1:(10 points)

A. Show the correctness of the following expressions (show as equivalence) without using truth table, and cite the rule

I. $(p \Rightarrow (q \vee \sim((p \vee q))) \vee (\sim p \wedge q) \equiv \sim(p \Rightarrow q) \vee p$

II. $(p \wedge(\sim(\sim p \vee q))) \vee ((p \wedge q) \equiv \sim(p \wedge q)$

B. Given the following definitions:

a: *"Ahmed's cat is alive"*
b: *" the box is open"*
c: *"the poison was released"*

I. Translate the given English sentences into logical notions:

1- Ahmed's cat is alive and the box is open
2- If the box is closed and the poison is released, then the cat is dead

II. Translate the following formal statement into English sentences.

1- ~c => a
2- (~b and c) -> ~a

C. Write a well founded formulae that specify the following arguments:

"if the value of the first element of the array M[1..N] is negative then the linked list P is empty, otherwise, the following j elements of the array contains positive integers (pointer values) of the indicated nodes of the linked list"

Question 2: (17 points)

Given the program segment in the following page:

A. Draw the flowgraph of the program (2 points)
B. Analyze the program (use path analysis) (4 points)
C. Drive the verification schema for the program (5 points)
D. Verify the program. (6 points)

```
Pre: {odd(wc) , bc > 1}
     while (wc + bc > 1) do
      begin
      white :=false; black := false;
      get(value)
      if (value = "w")
        then begin white:= true; wc :=wc -1 end
        else  begin black := true; bc := bc -1 end;
     get(value)
      if (value = "w")
        then begin white:= true; wc :=wc -1 end
        else  begin black := true; bc := bc -1; end
     if( black = white) then   wc := wc +1
                        else bc := bc+1
Inv:{ odd(wc)}
     End
postLoop:{wc=1 ∧ bc=0}
```

Question 3: *(16 points)*

Prove the partial correctness of the following: (verification schemas and proof of each)

A. *(3 points)*

Pre:{r ≥ π ∧ m > π/2}

x := r – m; y:= r+π/2; x:= x + y; y := Tan(x); x := x+y;

Post: { x > y ≥ 0 }

Where: Tan(x) = Sin(x)/Cos(x)

B. (Drive PostLoop) *(7 points)*

```
{Loop Invariant(i,q,v,w) : v = w *q +i  }

{Pre: v ≥ 0, w > 0 }
     q := 0;  i := v;
      while (w <= i) do
       begin
          i :=i - w
          q := q +1;
{Loop Invariant}
       end;
{PostLoop:  ???}
```

Institute of Statistical Studies and Research
Department of Computer and Information Sciences
CS601-Verification of Computer Programs
January, 2010 Time permitted: **Three Hours**

==========

Solve all questions. **Write Complete Answer for each question and Justify it**

==========

Question #1 *(10 Points)*

Given the following program segment:

Pre: x> 0, y>=0

```
1.   read(x, y);
2.   j :=1;
3.   i :=0;
4.   while (i < y) do
       begin
5.     j := j * x;
6.     i := i+1;
       end;
```

Invariant: J(x,i) ⇔ After iteration i, j contains the i^{th} power of x.

A. Draw the flowgraph of the program. *(1.5 points)*
B. Analyze the program (use path analysis). *(1.5 points)*
C. Prove by mathematical induction that ***J(x, i)*** is a loop invariant. *(3.5 points)*
D. Write the verification schema for this program segment. *(3.5 Points)*

Question #2 *(8 Points)*

A. Write a well founded formulae that specify the following argument, If here is an error, identify the type of error. *(2 Points)*

"If your patient is breathing, she is alive. Your patient is not breathing. Therefore, your patient is not alive."

B. Show whether the predicates in the following page are a tautology or not in the given interpretations: *(3 Points)*

$\forall \alpha \; \exists \beta \bullet \; \alpha = \beta + \pi/2 * \sqrt{-1}$

I. $\alpha \in$ rational numbers and $\beta \in$ irrational numbers
II. $\beta \in$ rational numbers, $\alpha \in$ imaginary numbers
III. α and β are real variables in a programming language.

C. Rewrite each of these statement so that negation are applied exclusively to predicate. *(3 Points)*

i. $\sim (\exists x \; \exists y \sim P(x,y) \wedge \forall x \; \sim \forall y \; Q(x,y))$

ii. $\sim \forall x(\exists y \; \forall z \sim P(x,y,z) \; \wedge \sim \exists z \; \forall y \; P(x,y,z))$

Question #3 *(15 Points)*

A. Drive the **wp()** for the following program segment and show if it is correct. State the initial values that would make ***{Pre} S {post}*** a valid predicate. *(5 Points)*

Pre:{True}
S:[

```
1    J := 0;
2    m1 := 0;
3    m2 := 1;
4    m3 := 7;
5    m1 := m1 + m3;
6    m2 := m2 + m3;
7    m3 := m3 + m2;
```

]

Post: $\{ m1 = j \wedge m2 = j + 1 \wedge m3 = 8 + j \}$

B. Prove the correctness of the following program segment, (verification schemas and proof): State the initial values that would make ***<Pre> S <post>*** a valid predicate. *(5 Points)*

< ***Pre:*** $r > 0, a \geq 0$ >
S:[

```
1  r := r *a;
2  t:= r - a;
3  If (t < a)
4     Then  t := t * r
5     Else  t := t –ln(a + r)
```

]

< ***Post:*** $t \geq 0$ >

C. Drive the Invariant {I}, then drive VCs for the following program segment: *(5 points)*

Pre: $\{z > 7, -5 < w < 0\}$

1 $while(z > w)$

2 $w = -w + 1;$

$\{I\}$

Post: $\{w > 0\}$

Question 4*: (15 points)*

Given the following annotated Program segment:

Pre: $\{N > 0\}$

1 $i := 1;$

2 $k := N-1;$

$\{INV(I,k,a)\}$

3 ***while*** $i \leq k$ ***do***

begin

4 ***if*** $(a[i] < 5)$

5 ***then*** $a[i] := 5$

6 $i := i + 1$

$\{INV(I,k,a)\}$

end

Post: $\{(\forall m \mid 1 \leq m \leq N \bullet a[m] >= 5)\}$

A. Draw the flowgraph. *(1 point)*
B. Analyze the program (use path analysis). *(2 points)*
C. Write the verification schema for this program segment. *(5 points)*
D. Given the following predicate, prove that it is an invariant. (5 *points*)

$Inv(I, k, a) \Leftrightarrow \{(\forall j \mid I > j \geq 1 \bullet a[j] \geq 5), I \leq k + 1\}$

E. What modification, if any, does ***{Inv}*** need to be *THE* Invariant?(*2 points*)

Question 5: *(14 points)*

Pre: { N > p > 0, x positive integer > 0}

```
1.   1:  I:=1;
2.   2:  while( I <= p) do
3.       begin
4.   3:    if (a[I] = x)
5.   4:      then  I :=p+1;
6.   5:    I := I +1;
7.       end;
8.   6:  if (I  =p+1) then
9.        begin
10.  7:     p := p+1;
11.  8:     a[p] := x;
12.       end;
13.      Exit
```

Post:{ $x \notin a_o \Rightarrow (p = p_o + 1 \wedge x = a[p]$}

A. Draw the flow graph of the given implementation. *(1.5 points)*
B. Develop test suites that guarantee that the loop will be executed exactly: 0 time, 1 time, 'p' times and 'x' times. *(1.5 points)*
C. State the Def-Use Chain for the variables a, i, p and x. *(2 points)*
D. Drive the test suite that covers the U-context at **8**, in terms of path (U-Context Constructor). *(4 points)*
E. Find the static slice for the criteria C_s= <8, a>. *(5 points)*

Question 6: *(8 points)*

Given the specification of a program, that calculates the wages for Part time hires. The minimum wage per hour, "*MWage*", is given, as well as, a disk file containing records for the hired individuals. Each record composed of the hire's name "*Hname*", and their corresponding working hours "*WH*".

Pre: MWage > 5, Hname ∈ alphabet letters, WH > 5
Post: {∀ Hname ∈ hired individual, ∃ wage > 25}

A. Provide a formal specification for the functionality of the program. *(2 points)*
B. Drive Input, output test suites for the program. *(2 points)*
C. Drive the domain and range test suites for the program. *(2 points)*
D. Drive a functionality test suite for the program. *(2 points)*

☺ GOOD LUCK

Institute of Statistical Studies and Research
Department of Computer and Information Sciences
Advanced Topics in Software Engineering – CS601
January 2012
Time permitted: Three Hours

Solve all questions. **Write Complete Answer for each question and Justify it.**

Question 1: (10 points)

Given the following formal specification of a given problem requirements:

Pre: $\{p > 0\}$

Post: $\{(\forall i \mid 1 \le I \le p \bullet a[I] \neq x) \Rightarrow (x = a[p+1])\}$

A. Derive a Black-Box test for the implemented specification. Clarify your answer by stating the various types/techniques and driving at least **two test suites** for each of them. (5 points)

B. What are the possible and potential faults that could be triggered by each of the test suites that you drove in answering (B)? (2 points)

C. Write a **well founded formulae** , that gives the approximate number of symbols in a positive integer. The integer is written in base 16. (3 points)

Question 2: (14 points)

Given the following program segment:

Pre: u > 0, v>=0

```
1.  read(u, v);
2.  p :=1;
3.  i :=0;
4.  while (i < u) do
      begin
5.      p := p * v;
6.      i := i+1;
    end;
```

Invariant: J(x,i) ⇔ After iteration i, p contains the i^{th} power of v.

Post: {POST}

A. Draw the Flow graph for the program. (2 points)

B. Analyze the program (use path analysis). (2 points)

C. State the verification schema and derive the VCs for this program segment. (6 points)

D. Verify/prove VC_0 and VC_i. (4 points)

Question 3: (12 points)

Given the program segment in the following page:

1. Prove by mathematical induction that P(k) is a loop invariant. (5 points)
2. Derive the traversal condition for the *exit from* the loop. (2 points)
3. State and Derive the verification schema for this program segment. (5 points)

```
Pre: {m>0}
    1     nodd := 0;  k :=1;
    2     while k ≤m do
            begin
    3       nodd :=  nodd + 2k;
    4       k: = k + 1;
            end
          k
{P(k) ⇔ Σ 2i = k *(k+1)}
         i=1
```

Given the following program segment, answer questions 4, 5 and 6.

```
Pre:{ {N, a} ∈ natural numbers }
    1     i := 1;
    2     k := N-1;
    3     while i ≤ k do
          begin
    4       if (a[i] < 9)
    5          then a[i] := 9 ;
    6     i := a[i]+1;
          end
Post:{(∀m /1 ≤ m ≤N • a[m] >= 9 )}
    7     writearray(a) // print the content of the array a
```

Question 4: (13.5 points)

A. Draw the flow graph. (2 points)
B. Analyze the program (use path analysis). (2 points)
C. State the life variables at instructions (**nodes) *2* and *5***. (2 points)
D. Derive the control scope for all control instructions. (3 points)
E. State the Def-Use Chain for the variables ***i, k*** and ***a***. (4.5 points)

Question 5: (9 points)

A. Assuming that *N > 2 and i= 1 (as given)*, what are the values of the array elements that guarantee that the loop will be executed exactly: 0 times, 1 time, 4 times and 9 times before exit the loop. (4 points)
B. Derive the traversal condition and the verification schema for the ***loop body***. (5 points)

Question 6: (11.5 points)

A. Derive the test suite that covers the U-context at 5, in terms of path (U-Context Constructor). (3.5 points)
B. Generate a test suite for the constructor in part (A). (2 points)
C. Find the static slice for the criteria C_s= <7, {a}>. (6 points)

☺ **Best of Luck**

Institute of Statistical Studies and Research
Department of Computer and Information Sciences
Advanced Topics in Software Engineering – CS601
September 2015
Time permitted: Three Hours

Question #1: *(20 points)*

A. Show that the following expressions with predicates P, Q and R. *(4 points)*

i. $P \Rightarrow Q$ is logically equivalent to $(\neg Q) \Rightarrow (\neg P)$.
ii. $P \Rightarrow (Q \Rightarrow R)$ is logically equivalent to $(P \wedge Q) \Rightarrow R$.
iii. $\forall x(P(x)) \wedge \exists x(Q(x))$ is logically equivalent to $\forall x(\exists y\ (P(x) \wedge Q(y)))$

B. Derive the following:

i. WP(`Pi:=22/7; C:= D-pi; B:= C-2*D; A:= sin(B+ Pi), {A=22/7}`) (5 points)
ii. WP(`if x > 0 then y:= a + x else y:= a - x, {y > a+x}`) (5 points)
iii. The ***Verification Conditions*** for the following program segment: (6 points)

```
Pre : {0  <> x ∧ y ≤ 0}
        W :=  y/x;
        z :=  w-y;
        if (w > z)
         then  w :=  w * x
         else  w :=  w * y ;
Post : {w ≥ 0}
```

Question 2: (16 points)

Derive **the Loop Invariant** for each of the following program modules:

A. *(8 points)*

```
read(N);
j :=1;
i :=0;
while (i < N) do
  begin
    j := 2*j ;
    i := i+1;
  end;
Loop invariant:  {??}
```

B. *(8 points)*

```
 Pre: { n > 0}
sum = 0;
for (i = 1; i <= n; i ++)
       sum += 2*I - 1 ;
 Loop invariant:  {??}
```

Question #3: *(20 points)*

Given the program segment in the following page:

A. Draw the flow graph of the program. *(3 points)*
B. Prove by mathematical induction that P(k) is a loop invariant. *(7 points)*
C. Derive the traversal condition for the *exit from* the loop. *(3 points)*
D. State and Derive the verification schema for this program segment. *(7 points)*

```
Pre: {m>0}

1    nodd := 0;
2    k :=1;
3    while k ≤ m do
           begin
4            nodd :=  nodd + 2k;
5            k: = k + 1;
           end
```

$$\{\text{Post: } \sum_{i=1}^{k} 2i = k*(k+1)\}$$

Question 4: *(14 points)*

Given the following program segment:

```
Pre: u > 0, v>=0
1.      read(u, v);
2.      p :=1;
3.      i :=0;
4.      while (i < u) do
            begin
5.            p := p * v;
6.            i := i+1;
            end;
```

Invariant: J(x,i) ⇔ After iteration i, p contains the i^{th} power of v.

Post: {POST}

A. Draw the Flow graph for the program. *(2 points)*
B. Analyze the program (use path analysis). *(2 points)*
C. State the verification schema and derive the VCs for this program segment. *(6 points)*
D. Verify/prove VC_0 and VC_i. *(4 points)*

CONSIDER THE FOLLOWING IMPLEMENTATION FOR Questions 5 and 6:

PRE: { N > p > 0, x > 0, a[] is a non empty array of integers}

```
1.   1:  I:=1;
2.   2:  while( I <= p) do
3.         begin
4.   3:     if (a[I] = x)
5.   4:       then I := p + 1;
6.   5:     I := I +1;
7.         end;
8.   6:  if (I = p+1) then
9.         begin
10.  7:     p := p+1;
11.  8:     a[p] := x;
12.        end;
13.      Exit
```

POST:{ $x \notin a_0 \Rightarrow (p = p_0 + 1 \wedge x = a[p]$}

Question 5: *(15 points)*

A. Draw the flow graph of the given implementation. *(3 points)*

B. Analyze the program (use path analysis). *(3 points)*

C. State the life variables at instruction **(node)** *7.* *(3 points)*

D. State the Def-Use Chain for the variables **i, p, x** and **a[]**. *(6 points)*

Question 6: *(15 points)*

A. Develop a test suite that guarantee that the loop will be executed exactly: 4 times, then exit the loop to 6 then directly exit the program. (3 points)

B. Drive the test suite that covers the U-context at 8, in terms of path (U-Contex Constructor). *(7 points)*

C. Find the processing nodes PN of the static slice for the criteria $C_s = \langle 8, a[] \rangle$. *(5 points)*

Best of Luck ☺

Cairo University	Cairo University – Institute of Statistical Studies and Researches		Cairo University
	Department: Computer Sciences		
	Academic Year: 2015/2016	Semester: Two	
	Date: June 5th.	Level: Pre-Master	

Course Title: Advanced Topics in Software Engineering	Course code: CS601	Time: 3 Hours	Exam marks: 100	# Exam. Sheets: 2
Exam. Instructions : Calculators and cell phones are not permitted				

Question #1: (16 points)

A. Show the correctness of the following expressions (show as equivalence) without using truth table, and cite the rule 4

1. $\sim(\sim(p \vee \sim r) \wedge q) \vee (\sim p \vee \sim q) \equiv p \vee \sim q \vee r$
2. $p \vee \sim((p \vee q) \wedge r \equiv p \vee \sim q \vee \sim r$
3. $(p \Rightarrow (q \vee \sim((p \vee q))) \vee (\sim p \wedge q) \equiv \sim(p \Rightarrow q) \vee p$
4. $(p \wedge (\sim(\sim p \vee q))) \vee ((p \wedge q) \equiv \sim(p \wedge q)$

B. Rewrite each of these statement so that negation are applied exclusively to predicate, i.e., no negation precedes a quantifier or an expression involving logical connectives)

1- $\sim \forall y \; \exists x \; P(x,y)$

2- $\sim \forall x(\exists y \; \forall z \; P(x,y,z) \wedge \exists z \; \forall y \; P(x,y,z))$

3- $\sim \exists z \; (\forall y \; (z \Rightarrow p(y) \cap \sim p(x) \Rightarrow z)$

4- $\sim \exists y \; \forall x \; (\exists z \; p(x+z) \Leftrightarrow p(x+y))$

Question #2: (12 points)

Prove by Mathematical induction the validity of the following loop:
(formulate the loop invariant, then show that it is a loop invariant)

PRE: $\{n \leq 1\}$

```
1     mut := z;
2     j := 2
3     while (j <= n)
        begin
4         mut := mut * x;
5         j := j + 1;
        end;
```

POST: $\{Mut = x^n\}$

Question #3: (20 points)

Derive the following:

(A) WP(*Pi:=22/7; C:= D+Pi/2; B:= C*Pi; A:= sin(B+2*Pi)*, $\{A=1/2\}$) (4 points)

(B) WP(*if (x = 0) then y:= a else y:=a+x*, $\{y=a+x\}$) (4 points)

(C) The ***Verification Conditions*** for each of the program segments:

C1:

```
pow(x,y):
Z=1; m=0;
While (m< y)do
 {z *= x; m += 1; }
return z
```

C2: Pre:{a <-5 ∨ a > 5}

```
if (a < 0)
   then c := -a
   else  c := a-1
```

Post:{ c ≤ |a| }

Question #4: (20 points)
Given the follwing program segment :

Pre: { n ≥ 0}

fact = 1;
for (i = 1; i <= n; i ++)
*Fact *= i;*

{ Loop invariant}: After the K^th iteration, fact = 1*2*...*k

A. Draw the flow graph of the program (2 points)
B. Analyze the program (use path analysis) (4 points)
C. Drive the verification schema for the program (5 points)
D. Verify the program. (6 points)

Question #5: (16 points)

Generate a black box test suites for a program that computes the roots X_1 and X_2 **of** a quadratic equation $ax^2 + bx + c$ given the coefficients *a, b* and *c*, where:

Preconditions: $\{a, b, c\} \in$ **real numbers**

Postconditions: $X_1, X_2 \in$ **irrational numbers**

Note that: $X_{1,2} = (-b \pm \sqrt{b^2 - 4ac})/2a$

Question #6: *(16 points)*

Given the following program segment in the next page:

A. Draw the flow graph of the given implementation.
B. State the Def-Use Chain for the variables i, j and y.
C. Drive the test suite that covers the U-context at **5, in** terms of path (U-Context Constructor).
D. Find the static slice for the criteria *C= <7,{j}>*

```
1.   1.   read(x, y);
2.   2.   j :=1;
3.   3.   i :=1;
4.   4.   while (i < y) do
5.        begin
6    5.    j := i * x;
7.   6.    i := i+1;
8.        end;
9.   7.   write (j);
```

---------- ☺ GOOD LUCK ☺ ----------