Answer 1. (B) Consistency in database systems refers to the requirement that any given database transaction must only change affected data in allowed ways, that is sum of x and y must not change.

Answer 2.

There are two possible executions: $T_1\ T_2$ and $T_2\ T_1$.

Case 1:

	A	B
initially	0	0
after T_1	0	1
after T_2	0	1

Consistency met: $A = 0 \vee B = 0 \equiv T \vee F = T$

Case 2:

	A	B
initially	0	0
after T_2	1	0
after T_1	1	0

Consistency met: $A = 0 \vee B = 0 \equiv F \vee T = T$

Answer 3. Yes, as the graph is acyclic. A possible schedule can be obtained from a topological sort- one schedule is T1 , T2 , T3 , T4 , T5

Answer 4. (a)T1 -> T3 -> T2 Explanation: T1 can complete before T2 and T3 as there is no conflict between Write(X) of T1 and the operations in T2 and T3 which occur before Write(X) of T1 in the above diagram. T3 should can complete before T2 as the Read(Y) of T3 doesn't conflict with Read(Y) of T2. Similarly, Write(X) of T3 doesn't conflict with Read(Y) and Write(Y) operations of T2. Another way to solve this question is to create a dependency graph and topologically sort the dependency graph. After topologically sorting, we can see the sequence T1, T3, T2

(b) No schedule is not recoverable, (c) No

Answer 5 !(10+5+6) / !10. !5.!6

Answer 6 Blind writes appear in any schedule that is view serializable but not conflict serializable

T1	T2	T3
Read(Q)		
	Write(Q)	
Write(Q)		
		Write(Q)

Answer 7

T1	T2
R(A)	
A=A-50	
W(A)	
	R(A)
	A=A+20
	W(A)
	Commit
R(B)	
B=B+50	
Commit	

Precedence graph T1 -> T2;

A recoverable schedule is one where, for each pair of transactions Ti and Tj such that Tj reads a data item previously written by Ti, the commit operation of Ti appears before the commit operation of Tj .
* Recoverable schedule = Roll backing of un committed transaction
* Non-recoverable schedule = Roll backing of committed transaction