1.
a)
P(Smart, Study, Pass)=P(Smart)×P(Study)×P(Pass | Smart, Study)

b)

P(Smart,Study,Pass)	smart		-smart	
	study	-study	study	-study
pass	.1083	.084	.072	.024
-pass	.006	.036	.048	.096

c)
P(Pass | No Study)=P(Pass | Smart, No Study)×P(Smart)+P(Pass | Not Smart, No Study)×P(Not Smart)

.6

d) From the JPT, compute the probability that a student did not study, given that they are smart but did not pass the test.

P(No Study | Smart, Not Pass)= P(Not Pass | Smart) / P(Not Pass | Smart, No Study)×P(No Study)

.9

e)
P(Pass | Smart)=P(Pass | Smart, Study)×P(Study)+P(Pass | Smart, No Study)×P(No Study)

.8

f)
P(Pass | Study)=P(Pass | Smart, Study)×P(Smart)+P(Pass | Not Smart, Study)×P(Not Smart)
0.705

2)
a)
P(Cold, Sneeze, Allergic, Scratches, Cat)= P(Cold)×P(Cat)×P(Allergic | Cold, Cat)×P(Sneeze | Cold, Allergic)×P(Scratches | Cat)

b) Use the equation above to calculate the joint probability that the person sneezes, but does not have a cold, has a cat, is allergic, and there are scratches on the furniture:
P(-cold,sneeze,allergic,scratches,cat) = ?

0.03591

c)
P(cat,-cold,sneeze,allergic,scratches)
/[P(cat,-cold,sneeze,allergic,scratches) +P(-cat,-cold,sneeze,allergic,scratches)]

P(cat,-cold,sneeze,allergic,scratches) = 0.00064
P(-cat,-cold,sneeze,allergic,scratches) =.031421
0.00064/(0.00064+.031421) = .019961

d)
P(scratches|cat)P(cat)/P(scratches)

0.5(0.05)/P(scratches)

e)
P(-cold, sneeze, allergic, scratches, cat): P(-cold)×P(cat)×P(allergic|-cold, cat)×P(sneeze|-cold, allergic)×P(scratches|cat)

P(-cold, sneeze, allergic, scratches, -cat): P(-cold)×P(-cat)×P(allergic|-cold, -cat)×P(sneeze|-cold, allergic)×P(scratches|-cat)

2 joint probabilities

3)
a)
start-car(c):
  pre-conds: at(c), has-key(c), charged-battery(c), has-gas(c)
  effects: car-running(c), ¬has-gas(c), at(c), has-key(c)
b)
∀c, s [At(c, s) ∧ HasKey(c, s) ∧ ChargedBattery(c, s) ∧ HasGas(c, s) →
    CarRunning(c, do(StartCar(c), s)) ∧ At(c, do(StartCar(c), s)) ∧
    HasKey(c, do(StartCar(c), s)) ∧ ¬HasGas(c, do(StartCar(c), s))]

c)
∀c1, c2, s [c1 ≠ c2 → (OutOfGas(c2, s) ↔ OutOfGas(c2, do(StartCar(c1), s)))]