

1. Translate the following into English

$$\exists x \bullet \text{class}(x) \wedge \forall y \bullet \text{student}(y) \wedge \text{lazy}(y) \wedge \text{takes}(y,x) \Rightarrow \text{fails}(y,x)$$

where

class(x) means *x is a class*.

student(x) means *x is a student*.

lazy(x) means *x (a person) is lazy*.

takes(x,y) means *x (a person) takes y (a class)*

fails(x,y) means *x (a person) fails y (a class)*.

2. Encode the following into the language of predicate logic.

Every customer receives a gift when he registers.

Let

customer(x) mean *x is a customer*.

gift(x) mean *x is a gift*.

registers(x) mean *x registers*.

receives(x,y) mean *x receives y*.

3. What does the following abstract program specify?

$$y : [\exists z \bullet \text{age}(z,x) \wedge z \geq 21, \\ \exists z \bullet \text{parent}(z,x) \wedge (\text{sister}(y,z) \vee \exists u \bullet \text{brother}(u,z) \wedge \text{wife}(y,u))]$$

where

age(x,y) means *x (a natural number) is the age of y (a person)*.

parent(x,y) means *x is a parent of y*.

sister(x,y) means *x is a sister of y*.

brother(x,y) means *x is a brother of y*.

wife(x,y) means *x is the wife of y*.

4. Write an abstract program that specifies the following. As long as the sum of *y* and *z* is no greater than *x*, increment both *y* and *z* by *x*. (*x* is read-only while *y* and *z* are read-write.)