DEMO 3 Mistakes:

1: Task

a) Some of you did not specify that “Employee”, “University”, “Course”, “Unit” are sub-classes of rdfs:Resource class as well as missed definition of rdfs:Resource as a class.
b) Do not forget to put all necessary prefix definitions in case you use those prefixes.
c) Some of you did not specify that classes are type of rdfs:Class (e.g. u:Employee rdf:type rdfs:Class)
d) Be sure that you do not make any misprints with prefixes (such as rdfs:Property instead of rdf:Property) because they are different URIs.
e) Good practice is to write names of properties and individuals from the small letter and names of the classes from the big ones. You may do not follow this practice, but you cannot mix small and big letters of already defined concepts (especially the concepts from other ontologies): rdfs:Domain and rdfs:Range do not exist.
f) Some of you mix up property definition (definition of the property domain and range) with utilization of the property to define relations between the individuals.

```
u:Course          a                        rdfs:Class.
u:Lecturer        rdfs:subClassOf          u:Employee ;
                  a                        rdfs:Class ;
                  u:teaches                u:Course ;
                  u:hasFirstName           xsd:string ;
                  u:hasSurName             xsd:string .
```

in this case property *u:teaches* is used to define relation between the instance (individual) *u:Lecturer* (instance of the “rdfs:Class” class) and another individual *u:Course* (instance of the “rdfs:Class” class). In this case the range of the *u:teaches* property is “rdfs:Class” and the domain is “rdfs:Class” as well.
In the task you suppose to define the property through its’ domain and range definition:

```
u:teaches         a             rdf:Property ;
                  rdfs:domain   u:Lecturer ;
                  rdfs:range    u:Course .
```

2: Task

The right answers are following:

Statement	***Correct***	***Incorrect***
z:eagle z:hasLatinName "Aquila chrysaetos"	**X**	
z:dolphin z:hasLatinName "15.3"^^xsd:float		**X**
z:eagle z:eats z:dolphin	**X**	
z:eagle z:hasWingSpan "4.3"^^xsd:int		**X**
z:homoSapiens z:hasFingers "5"^^xsd:int	**X**	
z:eagle z:hasFingers "3"^^xsd:int		**X**
z:eagle z:hasWingSpan "5000"^^xsd:float	**X**	
z:elephant z:hasFingers "8"^^xsd:int	**X**	

3: Task

Some of you have problem with the meaning of sub-class definition (you mix it with “part of” relation). The logic behind sub-class relation is following: if class A is a sub class of class B, then instances of the class A share the properties of the Class B as well as can have some other (more specific) properties that belong to class A (e.g. classes Car, motorbike are sub-classes of a class Vehicle. Class SportCar is a sub-class of the Car class). In case of “part of” relation, some entities are the parts of some other entity, and it does not mean that they share common properties (e.g. different rooms are parts of an apartment - not sub-classes of it. Legs and hands are parts of a human body - not sub-classes of it).

4,5,6: Task

a) individuals.rdf file supposes to contain definition of new individuals (instances of classes defined in ontology) and triples that shows some relations between those individuals via properties defined in the ontology.
b) It has been asked to create individuals.rdf file separately from the protégé tool. But some of you just took (copied) individuals created in protégé and removed unnecessary parts. Unfortunately for you, not all the parts, which should be removed, has been removed (in case you are cheating, please do this more cleanly).
c) Some of you did not use all the mentioned ways of class definition.
d) Some of you do not understand difference between “create individuals of the class” and “define class as an enumeration of the individuals”. It is not the same. Please go through lecture materials one more time.
e) Some of your ontologies were not successfully validated by reasoner. One of the main reasons is that you define some class (class X) as compliment of another class (class Y) and at the same time this class (class X) is a subclass of class Y (or its’ subclass).