OOP Review

D.1.1 Outline the general nature of an object.

1. Outline the two characteristics of an object.

D.1.2 Distinguish between an object(definition, template or class) and instantiation.

2. Distinguish between a class and an object.

3. What does it mean to instantiate an object?

D.1.3 Construct unified modelling language(UML) diagrams to represent object designs.

4. Construct an UML diagram from the following class definition.

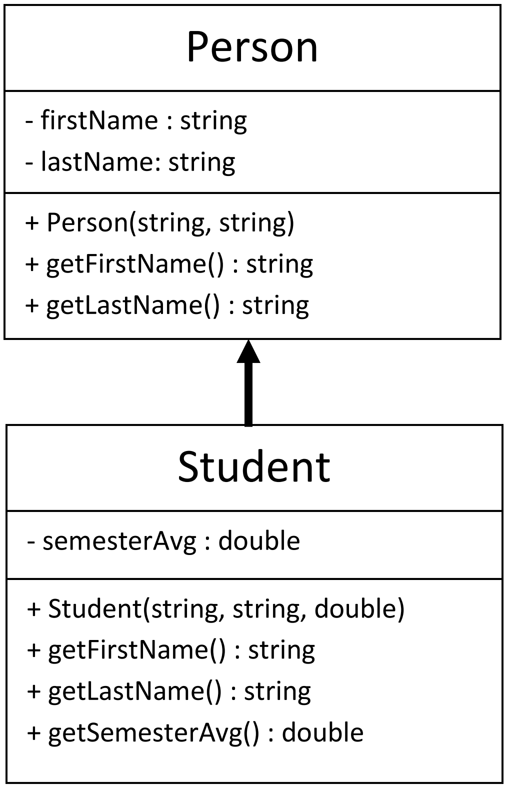
```
public class BaseballPlayer
{
     private String name;
     private int hits;
     private int atBats;

     public BaseballPlayer(String n, int h, int a)
     {  // code not shown  }
     public String getName()
     {  // code not shown  }
     public int getHits()
     {  // code not shown  }
     public int getAtBats()
     {  // code not shown  }
     public int calcBattingAvg()
     {  // code not shown  }
```

```
    public void setName(String n)
    {  // code not shown  }
    public void setHits(int h)
    {  // code not shown  }
    public void setAtBats(int a)
    {  // code not shown  }
}
```

D.1.4 Interpret UML diagrams

5. Write the two class definitions from the following UML diagram. Do not implement the methods.



D.1.5 Describe the process of decomposition into several related objects.

6. Describe class decomposition.

D.1.6 Describe the relationships between objects for a given problem.

7. Describe aggregation.

8. Describe inheritance.

D.1.7 Outline the need to reduce dependencies between objects in a given problem.

9. Outline the need to reduce dependencies between objects in a given problem.

D.1.8 Construct related objects for a given problem.

10. From the given examples tell whether the relationship between the two objects is aggregation or inheritance.

```
a. public class Business
   {
         private Employee employee;
         ...

b. public class Car extends Automobile
```

D.1.9 Explain the need for different data types to represent data items.

11. Why does Java provide 8 primitive data types?

D.1.10 Describe how data items can be passed to and from actions as parameters.

12. Describe the difference between a formal parameter and an actual parameter?

13. What is the scope of a formal parameter?

D.2.1 Define the term encapsulation.

14. Define the term encapsulation.

D.2.2 Define the term inheritance.

15. Define the term inheritance.

D.2.3 Define the term polymorphism.

16. Define the term polymorphism. How is polymorphism implemented in Java?

D.2.4 Explain the advantages of encapsulation.

17. Explain the advantages of encapsulation.

D.2.5 Explain the advantages of inheritance.

18. Explain the advantages of inheritance.

D.2.6 Explain the advantages of polymorphism.

19. Explain the advantages of polymorphism.

D.2.7 Describe the advantages of libraries of objects.

20. Describe the advantages of class libraries.

D.2.8 Describe the disadvantages of OOP.

21. Describe the disadvantages of OOP.

D.2.9 Discuss the use of programming teams.

22. Discuss the use of programming teams.

D.2.10 Explain the advantages of modularity in program development.

23. Explain the advantages of modularity in program development.

D.3.9 Discuss the features of modern programming languages that enable internationalization.

24. How does Unicode enable internationalization?

D.3.10 Discuss the ethical and moral obligations of programmers.

25. Should programmers be held accountable for the code they write? Why or why not?