1. Design a class named Person and its two subclasses named Student and Employee. Make Faculty and Staff subclasses of Employee. A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date-hired. A faculty member has office hours and a rank. A staff member has a title. Override the out method in Student, Employee, Faculty, and Staff classes to display the class name and the person's name.

FOR EXERCISES 2 TO 4, CONSIDER THE FOLLOWING CLASS:

```
public class Game{
        private String description;
        Game(  String newDescription )
        {
                set Description(  newDescription ) ;
        }
        publ ic String getDescription(  )
        {
                return description;
        }
        public void set Description(  String newDescription )
        { description = newDescription;
        }
        publ ic St r ing toString(  )
        {
                r e t u r n (  " d e s c r i p t i o n:  ”  +  d e s c r i p t i o n ) ;
        }
}
```

2. Write a class encapsulating a PC-based game, which inherits from Game. A PC-based game has the following additional attributes: the minimum megabytes of RAM needed to play the game, the number of megabytes needed on the hard drive to install the game, and the minimum GHz performance of the CPU. Code the constructor and the toString method of the new class. You also need to include a client class to test your code.

3. Write a class encapsulating a trivia game, which inherits from Game. A trivia game has the following additional attributes: the ultimate money prize, and the number of questions that must be answered to win the ultimate money. Code the constructor and the toString method of the new class. You also need to include a client class to test your code.

4. Write a class encapsulating a board game, which inherits from Game. A board game has the following additional attributes: the minimum number of players, the maximum number of players, and whether there is a time limit to finish the game.
Code the constructor and the toString method of the new class. You also need to include a client class to test your code.

For Exercises 5 to 6, consider the following class:

```
public class Store {
```

```
public final double SALESJAXJATE = 0.06;
private String name;
public Store( String newName )
{
        setName( newName );
}
public String getName( )
{
        return name;
}
```

5. Write a class encapsulating a web store, which inherits from Store. A web store has the following additional attributes: an Internet address and the programming language in which the website was written. Code the constructor and the toString method of the new class. You also need to include a client class to test your code.

6. Write a class encapsulating a restaurant, which inherits from Store. A restaurant has the following additional attributes: how many people are served every year and the average price per person. Code the constructor and the toString method of the new class; also code a method returning the average taxes per year. You also need to include a client class to test your code.