Problem Set

1. Create a base class Animal with a method makeSound() that prints "Some generic animal sound." Create two subclasses Dog and Cat that override the makeSound() method to print "Bark" and "Meow", respectively. In the main() method, create objects of Dog and Cat and call makeSound().
2. Create a superclass Shape with a method draw(). Create two subclasses Circle and Square that override the draw() method to print specific messages. Demonstrate polymorphism by creating a Shape reference and assigning it to both Circle and Square objects.
3. Define a class Vehicle with a method move(). Create a subclass Car that overrides move(). In the main() method, create an object of Car, upcast it to a Vehicle reference, and call the move() method to demonstrate polymorphism.
4. Create a class Employee with a method work(). Then, create two subclasses Manager and Developer, both overriding the work() method. In the main() method, create an array of Employee objects containing both Manager and Developer objects and call the work() method for each.
5. Create an interface Playable with a method play(). Create two classes Guitar and Piano that implement Playable and provide their own implementations of the play() method. Demonstrate polymorphism by calling the play() method on objects of both classes.
6. Create a class Calculator with two overloaded add() methods, one for integers and one for doubles. Create a subclass ScientificCalculator that overrides the add() method to perform a different calculation. Demonstrate method overloading and overriding in the main() method.
7. Create an abstract class Animal with an abstract method eat(). Create two subclasses Lion and Elephant that implement the eat() method differently. In the main() method, create objects of Lion and Elephant and call the eat() method to demonstrate runtime polymorphism.
8. Define a class Person with a method getDetails(). Create two subclasses Teacher and Student, both of which override getDetails(). Use polymorphism to create a Person reference and assign it to both Teacher and Student objects respectively.
9. Create a base class Device with a method start(). Create a subclass Phone that extends Device and overrides start(). Then create another subclass SmartPhone that extends Phone and overrides start(). Demonstrate polymorphism by creating objects of Device, Phone, and SmartPhone and calling start().
10. Create an interface Printer with a method print(). Create two classes LaserPrinter and InkjetPrinter that implement Printer and provide their own versions of print().

Demonstrate polymorphism by creating an array of Printer references and calling the print() method on both LaserPrinter and InkjetPrinter objects.

11. Create an abstract class Shape with an abstract method area(). Create an interface Drawable with a method draw(). Create two classes Rectangle and Circle that inherit from Shape and implement Drawable, providing specific implementations for both area() and draw(). Demonstrate polymorphism by using references of both Shape and Drawable types.
12. Create a class Person with overloaded constructors: one with no parameters and one with a String parameter. Create a subclass Employee that inherits from Person and overloads the constructor as well. Demonstrate constructor overloading and polymorphism by creating different types of Employee objects using both constructors.
13. Create a base class Payment with a method processPayment(). Create two subclasses CreditCardPayment and PayPalPayment, both overriding the processPayment() method. In the main() method, create a Payment reference and assign it to both types of objects to process payments using different methods.