

Inheritance Exercises

1. Using inheritance, create a program that prints out a different message depending on which child class is called.

a. Define an abstract class called Vehicle and give it appropriate properties/fields.

(Example:

int vehicleId = 1;

String registrationNumber = "ABC123"

String brand = "Volvo"

etc...)

b. Define three child classes: Car, Motorcycle and Truck. These classes should all extend the Vehicle class.

In addition, all three classes should have properties/fields unique to them.

(Example:

Car - numberOfDoors, numberOfPassengers...

Truck - maxSpeed, maxWeight...

Motorcycle - motorcycleType...)

c. Create an abstract method in the Vehicle class called drive().

Next, override the drive() method in all child classes and give it a different implementation depending on type of vehicle.

(Example:

Car - drive() { System.out.println("Car is driving...") }

)

d. Finally, make instances of all the child classes and call the drive() method from each class.

Assuming your implementation of the drive() method differs in each child class, you should get three different print outs.

2. Create a program that demonstrates additional features of inheritance.

a. Define a class called Book and give it the following fields:

title,author,category(enum),year and pages. Add getters and setters.

b. Create a default constructor that sets the pages field to 100.

Next, create an additional constructor that allows manual setting of the pages field.

c. Create category(enum) and give it three different values:

CHILD,ADULT and NOT_LOANABLE.

d. Define three child classes: ChildBook ,YoungAdultBook

and ReferenceBook. All classes should extend the Book class.

e. When creating a new ChildBook the category(enum) field should be set to CHILD. (Example: Category.CHILD). Do the same for YoungAdultBook, it's category(enum) field should be set to ADULT. Lastly, ReferenceBook's category(enum) should be set to NOT_LOANABLE.

f. Create an interface called Loanable. Give this interface a void method with no input parameters and call it loan().

g. A ReferenceBook can not be borrowed. Implement the Loanable interface only in the book classes that are able to be borrowed. Next, override the loan() method, and give it a message to print out.

3. Continue working with the program you created in previous Exercise.

a. Instantiate three books, one for each type of book, and store them in an array of type Book.

b. Iterate through the array and only print out the books that are instanceof the Loanable interface.

c. Iterate through the array and only print out the book that is not loanable.

d. Iterate through the array and only print out the book that has the category CHILD.

e. Iterate through the array and only print out the book that has the category ADULT.