Exercise 0.1:

Write a program that prints the numbers from 1 to 10 using a for loop

Exercise 0.2:

Write a program with a function called add, which takes 2 intergers and returns their sum

From main, call add(2,3) and print the returned value

Exercise 0.3:

Write a class called Book

Give the class a data member called number of pages (int)

in main create 2 books

set one book to have 20 pages, and the other to have 30 pages

print the values to check your code

Exercise 0.3b:

Create functions setNumberOfPages and getNumberOfPages inside the class book and use them from main

Exercise 0.4: ( Objects and pass by value )

Create a function in the main class ( class example ) that is called createLArgeBook.

Make the function create a new book with 100 pages and return it

Call the function from main and check that main received a book with 100 pages.

Exercise 0.5:

Create a function in the main class called add1page that accepts a book param

Make th function increment the book’s number of pages by 1

Call the function from main and check that it works

Exercise 0.6:

Change the Book class to have a constructor that takes the number of pages as a parameter

change main accordingly

Exercise 0.7:

Create a function called increaseNumbers that accepts 3 numbers, increases all of them by 10, and returns all 3 numbers to main

Arrays:

Excercise 1:

Create an array of 5 integers, fill them with values

Write a function sum , which receives the array as a parameter and returns the sum of the elements

Optional Advanced: Write a function min which finds the smallest element in the array

EXercise 1.2:

Create an array of 5 books, each with a different number of pages

Write a function sum that takes the array and returns the total count of all pages

Exercise 1.3:

Create a 2 Dimensional array of doubles (3X2) and fill it with values

Write a function that for each row , prints the sum of the numbers in the row

Arrays and Constructors:

Excercise 2:

Create an array that can hold 5 books

Write a function fillBooks that receives the array as a parameter and puts 5 new books with titles in the array

Write a function listBooks which receives the array as a parameter and and prints the titles of all the books

Excercise 2b:

Add a constructor to class Book that takes a String title argument

Encapsulation:

Excercise 3:

Create a class book with a field called title

Make the field private, and create a public setTitle function, and getTitle function

From main, create some books, and set their titles, and then print their titles, and test your code

Now change your code so that the setTitle function always makes the title upper case (e.g. WAR AND PEACE ), no matter what was sent in

Packages:

Excercise 4:

In your store excercise, put Book in the package storeProgram.storeItems

Put the class Store in the package storeProgram itself ( not package storeProgram.storeItems )

Inheritance:

Excercise 5:

Make a class Person with String name

Make a class Employee with name and int employeeNumber

in Class Example Make a static function void setPersonName( String firstName, String lastName, Person person )

which takes first name, last name, combines them e.g. "Gonen Israeli" and sets the person name to the combined name

Make sure your function works for employees and regular persons without copy paste

Inheritance and Polymorphism

Excercise 6:

• Create a Store class that will hold CDs and Books.

• Create methods for adding CDs and Books to the Store. Create methods to retrieve them from the Store.

• Make CD and Book

extend Item.

• Item will have name and price, and a printInfo function

• Create a CD class that adds a band member

• Create a Book class add an author member.

• Change the Store to have only one array member ( for books and cds )

• Change the client to work with items.

\* In the Store class, Add a function printAllItems to loop through the array and print all books and cds

Statics

Excercise 7:

Make a class Account that has a member accountNumber

Make accountNumber auto increment, so that the first Account object will get Account number 1, the second will get 2 etc.

Every time a new Account object is created with New, it should automatically get the next account number

Excercise 8: (Optional Advanced)

Create a class Log with a static function addToLog

Everytime an account is created, it should automatically add to the log "Account #... is being created"

Every time someone writes to the Log, it should add to the same Log message

Add a static function: String getLogMessage which will get the final combined log message

"Account 1 being created. Account 2 being created..."

Abstract Class

Excercise 9:

Create an abstract class Shape with an abstract draw function , but a fully implemented move(int deltaX, int deltaY ) function

Create class Circle and Rectangle which extend the abstract class and override draw

draw should print something like "Circle at xPos=5 yPos= 10"

Write in class Example a static function moveAndDraw which accepts a shape as a parameter , moves it 1 on x, 1 on y, and then draws it

Test your code

Interfaces

Excercise 10:

Turn your abstract class Shape into an interface

What changes do you need to make?

Excercise 11:

Add a new interface called HasCorners which has 1 function int getNumberOfCorners()

Make Rectangle only implement the interface ( Circle does not have corners )

Add a new static function void printCornerInformation which accepts a variable of type HasCorners, and nicely prints the number of corners

Check that your printCornerInformation works for Rectangle, but not for Circle

The Object Class

Exercise 12:

Create a class Ball that has fields int xPos, yPos

Create a toString function that returns a string "Ball: xPos=... yPos=..."

Create instances of ball, and print them using System.out.println

Create an equals function for your ball

Make sure that two balls with differnt coords are not equal, but balls with the same x,y coord are equal

Nested Class

Exercise 13:

Create a class Person with a data member String name

Create a class ID with two data members : int prefix, int postfix ( for id's like 555-444)

Make ID a nested class inside person. Create a member of type ID inside Person.

In main create a Person, give it a name and id. Print the person to see that all is OK