Storming Robots

Engineering For Kids!

JAVA – AP CS – EXERCISES

Last Update: August 8th, 2018

CONTENTS

About Standard Input / Output 3

Practice Problems I – The Basics 4

Practice Problems II – File I/o, String and basic Wrapper Classes 5

Ex1: 5

Ex2: exercise on File I/O 5

Ex3: About Wrapper Classes 6

Ex4: Write your own Class 10

Ex 5: Upcasting / Downcasting 11

Practice Problems III – Super Class, Sub-class, Inheritance and polymorphism 15

Ex 1- Implement your own Circle Class, Cylinder subclass, and Rectangle subclass 15

Ex2 – combine them all 16

Practice Problems IV – Arrays vs. ArrayList 17

Practice Problem V – Recursion 21

Practice Problems V – Sorting, Search Algorithms 22

About Standard Input / Output

```
//Calculate Area of a circle
import java.util.Scanner;

public class Area {

    public static void main (String[] args) {
        Scanner in= new Scanner(System.in);

        System.out.println("Enter Circle Diamenter:   ");
        float dia = in.nextFloat();
        float area = dia * dia * (float)Math.PI;

        System.out.println("\n Area of a circle with " + dia +
                " diameter = " + area);

        System.out.println(" ... or " + String.format("%.2f", area));
    }
}
```

```
import java.util.Scanner;

public class Temperature {

    public static void main (String[] args) {
        Scanner in= new Scanner(System.in);

        System.out.println("Enter Fahrenheit:   ");
        float fah = in.nextFloat();
        float cel = (fah -32.0f)  * 5.0f / 9.0f;

        System.out.println("\n F(" + fah + ") + = (C) " + cel);
    }
}
```

```
import java.util.Scanner;

public class Hyp {

    public static void main (String[] args) {
        Scanner in= new Scanner(System.in);

        System.out.println("Enter height and width:   ");
        int height = in.nextInt();
        int width = in.nextInt();

        System.out.println("\n h=" + height + "   width= " + width +
            ".  Hypotenuse = " + Math.sqrt(height*height + width*width)) ;
    }
}
```

PRACTICE PROBLEMS I – THE BASICS

1) Review the samples above, and write your own.
2) Create a program to ask user to enter a number as Fahrenheit. Your program will convert this to Celsius. Display the answer in 2 forms: integer value and floating number with 2 decimal places form.

   Celsius = (Fahrenheit – 32) * 5/ 9

   e.g. F = 75 C = 23 or 23.88

3) Then, create another program to ask user to enter Celsius. Your program will convert this to Fahrenheit.

4) Write a program to ask user to provide the radius of a circle, calculate the circumference and area display all values.

5) Plotting the "*" triangle. Write a function that outputs a right triangle of height and width n. If n = 6, the triangle will look like the image on the right.

   Tip: If you have difficulty in figuring out the loop structure, hardcode the layout of 6 levels, find the pattern, and convert to loop structure.

```
     *
    **
   ***
  ****
 *****
******
```

6) You borrow $1000. Your annual interest rate is 12%. You pay back $100 a month. If it is based on compound interest rate, how much do you still own after 4 months? The display should be something like this:

Output should look like this:

```
                  Interest at the      total owed at the
                  end of the month     end of a month
$ 1,000.00        $   10.00            $1,010.00          $ 910.00
$   910.00        $    9.10            $919.10            $ 819.10
$   819.10        $    8.19            $827.29            $ 727.29
$   727.29        $    7.27            $734.56            $ 634.56
```

7) Write a program to perform factorial of 2 to 6. Display the result backward as shown on the right. (note: your program must do the calculation, not hardcode the result.

```
6 !=        720
5 !=        120
4 !=        24
3 !=        6
2 !=        2
```

PRACTICE PROBLEMS II – FILE I/O, STRING AND BASIC WRAPPER CLASSES

Ex1:

Work on the exercises wrapper classes chapter in the baron book

Ex2: EXERCISE ON FILE I/O

1) Replace all the words “John” with “Benjamin” in a file.
   Steps:
   a. You should create a text file which have multiple “John” in it.
   b. Ask user to give you a file name.
   c. Ask user which word to replace and with what
   d. After that, your code should create an output file with the same file name but change the the extension to “.out” instead.
   e. Read in the content from an input file.
   f. Replace the words “John” with Benjamin.
   g. Write the replaced content into this *.out file.

   For example:

Enter the input file name: test.in What to replace: John Change it to : Benjamin	“test.in” is user’s input during run time. Your code should create an output file name as “test.out”

2) Create your own text file containing N number of words. Write a Java program to read in all words and produce the count of anagrams and what they are.

e.g. The text file contains:

Debit card Bad credit XXX Dormitory Dirty Room Astronomer Moon starrer School master The classroom I am Lord Voldemort Tom Marvolo Riddle	Your output: 5 sets of anagrams

EX3: ABOUT WRAPPER CLASSES

1) Work on the Wrapper Classes Chapter in the Barron Book.

2) Change MaxVariablesDemo to show minimum values instead of maximum values. You can delete all code related to the variables aChar and aBoolean. What is the output?

3) Create a program that reads an unspecified number of integer arguments from the command line and adds them together. For example, suppose that you enter the following:

```
java  yourAdderCode  10 20 30 40
```

3-a) The program should display 100 and then exit.
3-b) Generate an exception and exit if user enters only one argument.

4) Create a program that is similar to the previous one but has the following differences:
   - Instead of reading integer arguments, it reads floating-point arguments.
   - It displays the sum of the arguments, using exactly two digits to the right of the decimal point.

   For example, suppose that you enter the following:

```
java yourAdderFloat  1  1e2  3.0  4.754
```

   The program would display: 108.75.

5) What is the initial capacity of the following string builder?

```
String sb = new String("Able was I ere I saw Elba.");
```

6) Consider the following string:

```
String hannah = "Did Hannah see bees? Hannah did.";
```

   a. What is the value displayed by the expression `hannah.length()`?
   b. What is the value returned by the method call `hannah.charAt(12)`?
   c. Write an expression that refers to the letter `b` in the string referred to by `hannah`.

7) How long is the string returned by the following expression? What is the string?

```
"Was it a car or a cat I saw?".substring(9, 12)
```

8) In the following program, called `ComputeResult`, what is the value of `result` after each numbered line executes?

```
public class ComputeResult {
    public static void main(String[] args) {
        String original = "software";
        StringBuilder result = new StringBuilder("hi");
        int index = original.indexOf('a');

/*1*/   result.setCharAt(0, original.charAt(0));
/*2*/   result.setCharAt(1, original.charAt(original.length()-1));
/*3*/   result.insert(1, original.charAt(4));
/*4*/   result.append(original.substring(1,4));
/*5*/   result.insert(3, (original.substring(index, index+2) + " "));

        System.out.println(result);
    }
}
```

9) Show two ways to concatenate the following two strings together to get the string `"Hi, mom."`:

```
String hi = "Hi, ";
String mom = "mom.";
```

10) Write a program that extracts your initials from your full name and displays them.

11) An anagram is a word or a phrase made by transposing the letters of another word or phrase; for example,

"parliament" is an anagram of "partial men,"
"software" is an anagram of "swear oft."
“masters” is an anagram of “streams”

Write a program that figures out whether one string is an anagram of another string. The program should ignore white space and punctuation.

12) Review Questions:

1. he term "instance variable" is another name for ___.
2. The term "class variable" is another name for ___.
3. A local variable stores temporary state; it is declared inside a ___.
4. A variable declared within the opening and closing parenthesis of a method signature is called a ____.
5. What are the eight primitive data types supported by the Java programming language?
6. Character strings are represented by the class ___.
7. An ___ is a container object that holds a fixed number of values of a single type
8. Consider the following code snippet.

```
arrayOfInts[j] > arrayOfInts[j+1]
```

Which operators does the code contain?

9. Consider the following code snippet.

```
int i = 10;
int n = i++%5;
```

   a) What are the values of `i` and `n` after the code is executed?
   b) What are the final values of `i` and `n` if instead of using the postfix increment operator (`i++`), you use the prefix version (`++i`))?

10. To invert (negate) the value of a `Boolean expression`, which operator would you use?
11. Which operator is used to compare two values, `=` or `==` ?
12. Explain the following code sample:

```
result = someCondition ? value1 : value2;
```

13. What is the difference between **switch** and if-then-else.
14. What is the difference between **do {…} while(..);** statement vs **while(…) {…}** structure.

13) More difficult ones:

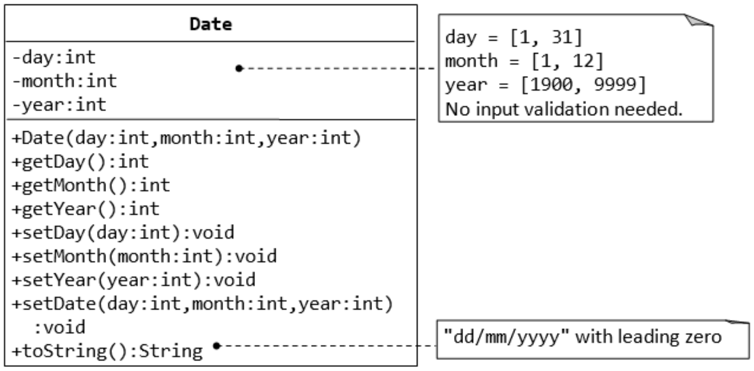
Extraction from Java Certification Exam published by Kalpanar.files.wordpress.com.

ANSWERS TO REVIEW QUESTIONS

1. The term "instance variable" is another name for **non-static field**.
2. The term "class variable" is another name for **static field**.
3. A local variable stores temporary state; it is declared inside a **method**.
4. A variable declared within the opening and closing parenthesis of a method is called a **parameter**.
5. What are the eight primitive data types supported by the Java programming language? **byte, short, int, long, float, double, boolean, char**
6. Character strings are represented by the class **java.lang.String**.
7. An **array** is a container object that holds a fixed number of values of a single type.
8. 1: `>`, `+`
9.
    a. `i` is 11, and `n` is 0.
    b. `i` is 11, and `n` is 1.
10. The logical complement operator "!".
11. The `==` operator is used for comparison, and `=` is used for assignment.
12. This code should be read as: "If `someCondition` is `true`, assign the value of `value1` to `result`. Otherwise, assign the value of `value2` to `result`."

EX4: WRITE YOUR OWN CLASS

1) Create the following Date Class.



```
Date
-day:int
-month:int
-year:int
+Date(day:int,month:int,year:int)
+getDay():int
+getMonth():int
+getYear():int
+setDay(day:int):void
+setMonth(month:int):void
+setYear(year:int):void
+setDate(day:int,month:int,year:int)
  :void
+toString():String
```

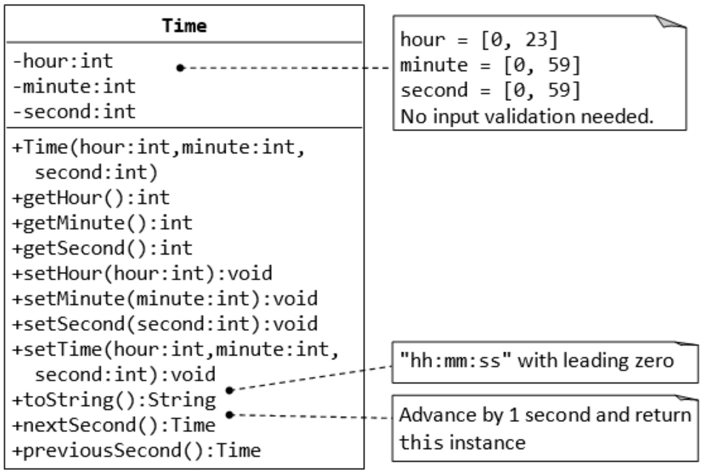
day = [1, 31]
month = [1, 12]
year = [1900, 9999]
No input validation needed.

"dd/mm/yyyy" with leading zero

After that, write a simple main class to create an Date object.

Then, use some of the methods to extract value which you created thru the constructor.

2) Create the following Time Class.



```
Time
-hour:int
-minute:int
-second:int
+Time(hour:int,minute:int,
  second:int)
+getHour():int
+getMinute():int
+getSecond():int
+setHour(hour:int):void
+setMinute(minute:int):void
+setSecond(second:int):void
+setTime(hour:int,minute:int,
  second:int):void
+toString():String
+nextSecond():Time
+previousSecond():Time
```

hour = [0, 23]
minute = [0, 59]
second = [0, 59]
No input validation needed.

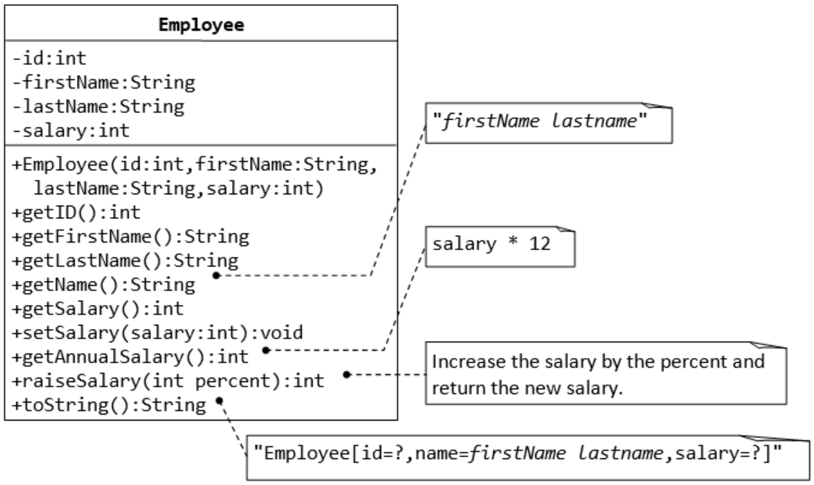
"hh:mm:ss" with leading zero

Advance by 1 second and return this instance

After that, write a simple main class to create an Time object.

Then, use some of the methods to extract value which you created thru the constructor.

3) Create an Employee class .. (a fully qualified class which you can instantiate)


Employee
-id:int
-firstName:String
-lastName:String
-salary:int
+Employee(id:int,firstName:String,
lastName:String,salary:int)
+getID():int
+getFirstName():String
+getLastName():String
+getName():String
+getSalary():int
+setSalary(salary:int):void
+getAnnualSalary():int
+raiseSalary(int percent):int
+toString():String
"firstName lastname"
salary * 12
Increase the salary by the percent and return the new salary.
"Employee[id=?,name=firstName lastname,salary=?]"

4) Create an Employee abstract class. And two fully qualified subclasses "manager" and "software developer" which will inherit Employee.

After that, write a simple main class to create two objects – one manager and one software developer.

Then, use some of the methods to extract value which you created thru the constructor.

Ex 5: Upcasting / Downcasting

1) What happens when the following program is compiled and run?

```
public class Employee {
  int z = 3;
  int doWork(){
    return z;
  }
}

public class Administrator extends Employee {

  int z = 8;

  int doWork(){
    return z;
  }

  public static void main(String[] args){
    Employee Employee = new Administrator();
    Administrator Administrator = (Administrator) Employee;
    System.out.print("-x" + Employee.doWork() + "-x" + Employee.z);
    System.out.print("-x" + Administrator.doWork() + "-x" + Administrator.z);
  }
}
```

2) What happens when the following is run?

```
public class Person {

  protected int i = 4;

  public int method(){
    return i * i;
  }
}

public class Student extends Person {

  int i = 3;

  public int method(){
    return i * i;
  }
  public static void main(String[] args){
    Person s = new Student();
    System.out.print("-x" + s.method());
    System.out.print("-x" + s.i);
  }
}
```

3) What happens when you compile and run the following:

```
public  class  Person {

  protected  int  i = 3;

  public  Person() {
    i ++ ;
    System.out.print("-x"  + method(5));
  }
  public  int  method(){
    return  i + i;
  }
  public  int  method(int  i){
    return  method() + i;
  }
}

public  class  Student extends  Person {

  int  i = 4;

  public  int  method(){
    return  i * i + super.i;
  }
  public  static  void  main(String[] args){
    Person Person = new  Person();
    Person Student = new  Student();
  }
}
```

4) What happens to the following code:

```
public class Person {
  protected int x = 5;

  public Person() {
    x ++ ;
    System.out.print("-" + method(7,3));
  }

  public int method(int i){
    return x + i - 4;
  }

  public int method(int i, int i2){
    return method(i) + i - i2;
  }
}

public class Student extends Person {
  int x = 3;
  public int method(int i){
    return (x * 2) + (i + super.x);
  }
  public static void main(String[] args){
    Person s = new Student();
    System.out.print("-" + s.x);
    Person sup = new Person();
  }
}
```

5) What happens with the following code:

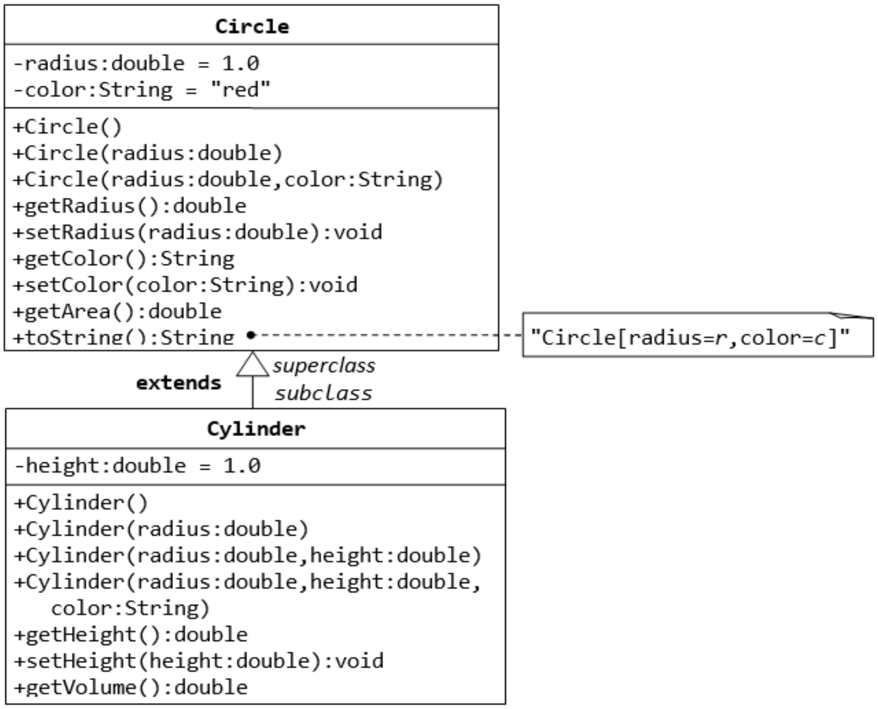
```
public class Person {
   protected int x = 5;

  public Person() {
    this(4);
    System.out.print("-" + method(7,3,2));
  }
  public Person(int i) {
    System.out.print("-" + method(i,2));
    x += 3;
  }
  public int method(int i, int i2){
    return i - i2 + x;
  }
  public int method(int i, int i2, int i3){
    return(i + i2 - i3 + x);
  }
}

public class Student extends Person {

  int x = 3;
  public int method(int x, int x2){
    return (this.x * 2) + (x + x2 + super.x);
  }
  public static void main(String[] args){
    Person s = new Student();

          }
}
```

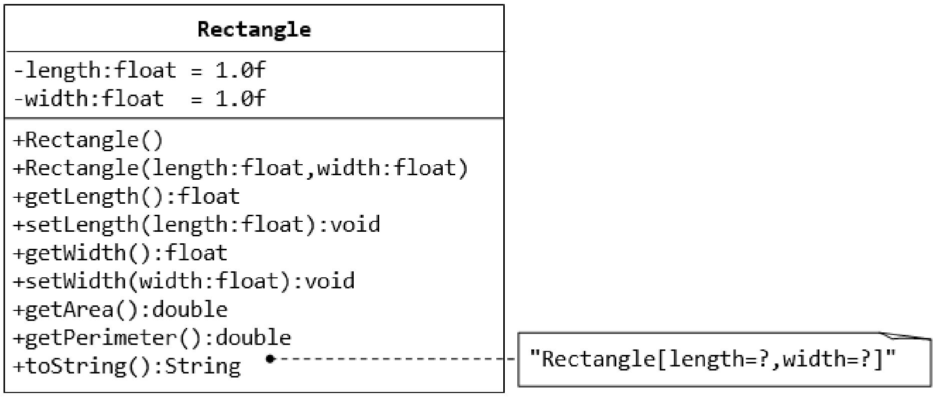
What if I do ?

```
a-   Student s = new Person();
b-   Student s = (Student) new Person();
```

PRACTICE PROBLEMS III – SUPER CLASS, SUB-CLASS, INHERITANCE AND POLYMORPHISM

EX 1- IMPLEMENT YOUR OWN CIRCLE CLASS, CYLINDER SUBCLASS, AND RECTANGLE SUBCLASS.





Ex2 – combine ***THEM ALL***

15. Create an Interface GeometryInterface Class.
16. Then, create an abstract Class Shapes to implement it.
17. Then, Circle class you did in the past exercise should inherit the new Abstract Shape class.

Let's start with…

```
interface GeometryInterfaces {
            // constant declarations, if any
            …
            void draw();
  void resize();
  void findArea();
  void findPerimeter();
}

abstract class AbsShapes  implements GeometryInterfaces{
   int x, y;
   ...
   void moveTo(int newX, int newY) {
      // this can be method implementation here.
   }
   Void resize();
              {
      // must implement it as it is not an abstract method.
   }
   abstract void draw();   // as it is an abstract method, you do not have to implement
   abstract void resize();
}

class Circle extends GraphicObject {
   void draw() {
      ...
   }
   void resize() {
      ...
   }
}
```

Practice Problems IV – Arrays vs. ArrayList

Review : https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html

Ex1 - Answer the multiple choice, write your own ArrayList to check out the answer.

1. Which of the following statements is NOT true about ArrayLists?

   a. ArrayLists are slightly faster than arrays.
   b. ArrayLists can store elements of different types.
   c. ArrayLists can increase in size to store more elements.
   d. ArrayLists have methods to manage their content.

2. Given the following code fragment:

   ```
   ArrayList<String> myArrayList = new ArrayList<String>( );
   myArrayList.add("One");
   myArrayList.add("Two");
   myArrayList.add("Three");
   myArrayList.add("Four");
   ```

3. Which of the following expressions will modify myArrayList so it looks like: One; Two; Four

   a. myArrayList.remove (myArrayList.get(3));
   b. myArrayList.remove (myArrayList.indexOf("Three"));
   c. myArrayList.remove (Three);
   d. myArrayList.remove (myArrayList.get(2));

4. Which of the following expressions will modify myArrayList so it looks like: One; Two, Three; Five

   a. myArrayList[3] = "Five"
   b. myArrayList[4] = "Five"
   c. myArrayList.set (myArrayList.indexOf("Four"),
   d. myArrayList.set (myArrayList.indexOf("Five"),

5. Given the following code fragment:

   ```
   ArrayList<Integer> myArrayList = new
   ArrayList<Integer>( );
   myArrayList.add(1);
   myArrayList.add(3);
   myArrayList.add(7);
   ```

6. Which of the following expressions will modify myArrayList so t looks like: 1 3 5 7

   a. myArrayList.add (5);
   b. myArrayList.add (2, 5);
   c. myArrayList.add (4, 5);
   d. myArrayList.add (3, 5);

EX3 – IMPLEMENT THIS ARRAYLIST AS ARRAY TYPE

Ask user to enter at least 5 different programming languages. You program should then display them in both the order they are entered and alphabetical order.   If the name entered is empty (spaces), eliminate it.

e.g.

```
Enter 5 different programming languages (with ; as the delimiter)

Python; C++  ; Java; PHP ;          C# ;      ;Perl
```

Output

```
Non-sorted:  Python; C++; Java; PHP; C# ; Perl
Sorted: C# ; C++; Java; PHP ; Python; Perl
```

EX4 - CREATE CLASS LISTNAME

In main() :

Create an ArrayList<String> students

Add 4 students to the ArrayList:

- Amy, BillyBob, CoolJoe, DennisTheMance
- Add them to the ArrayList directly

Write method to print elements in the ArrayList and its size

Write printArrayzlist() method with ArrayList as the argument

Your output should be:

```
Amy

BillyBob

CoolJoe

DennisTheMance

There are 4 names
```

Ex5 – Simple Anagram

Write a program to identify if 2 words are anagram.

Ex6 – find sets of Anagrams

Write a program to identify # of sets of anagrams in a list of 20 words, display which set each word is associated with.

Eg.
Input:

```
abc     cab     fib     biff     diff     ffid     bif     ifab     abfi     bac     ffib
```

Output:

```
abc     1
cab     1
afib    2
biff    3
diff    4
ffib    3
bif     5
ifab    2
abfi    2
bac     1
ffib    3
```

Ex7 - Write a program that reads a text file

Specified by:

- 1st argument : File name to read .
- 2nd argument: given the line # which allows your program will display that # of lines

Hint: To determine the number of lines in the file, use `java.io.File.length` to obtain the size of the file, then divide by an assumed size of an average line.

Sample implementation from docs.oracle.com. You should write your own.

```
import java.util.*;
import java.io.*;

public class FileList {
    public static void main(String[] args) {
        final int assumedLineLength = 50;
        File file = new File(args[0]);
        List<String> fileList =
            new ArrayList<String>((int)(file.length() /  assumedLineLength) * 2);
        BufferedReader reader = null;
        int lineCount = 0;
        try {
            reader = new BufferedReader(new FileReader(file));
            for (String line = reader.readLine(); line != null;
                    line = reader.readLine()) {
                fileList.add(line);
                lineCount++;
            }
        } catch (IOException e) {
            System.err.format("Could not read %s: %s%n", file, e);
            System.exit(1);
        } finally {
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {}
            }
        }
        int lines = Integer.parseInt(args[1]);
        for (int i = 0; i < lines; i++) {
            System.out.format("%d: %s%n", i, fileList.get(random.nextInt(lineCount - 1)));
        }
    }
}
```

PRACTICE PROBLEM V – RECURSION

Do the simple ones first from Baron's Book

From http://learn.stormingrobots.com. – find Recursion. Instead of using C/C++, use Java.
- note: this set of exercises is beyond AP CS required.

PRACTICE PROBLEMS V – SORTING, SEARCH ALGORITHMS

From Baron's Book:

- must understand the flow of the mergesort algorithm.
- Implement your own insertion sort and selection sort.

For advanced : Write your own qsort sorting implementation.