Array Worksheet Name:

Assume that the Book class below is implemented except for the method bestSeller. Use it to answer the questions.

public class Book

{

private String name;

private double price;

int numberSold;

public Book(String n, String p, int n){}

public double getPrice(){}

public String getName(){}

public void setName(String a){}

public void setPrice(double p){}

**public static Book bestSeller(Book[] books)**

**{//to be implemented in b)}**

}

1)

a)Write the static method mostExpensive in a driver class which accepts an array of Books and return the price of the most expensive book.

public static double mostExpensive(Book[] a){

double most=a[0].getPrice();

for(int i=0;i<a.length;i++){

if(a[i].getPrice()>most)

most=a[i].getPrice();

}

return most;

}

b) Implement the static method bestSeller in the Book class, which accepts an array of Books and return the best selling Book.

public static Book bestseller(Book[] books){

int index=0;

for(int i=1;i<books.length;i++){

if(books[i].numberSold>books[index].numberSold)

index=i;

}

return books[index];

}

2)Complete the following class.

public class Library{

private String libraryName;

private Book[] list;

public Library(String name, Book[] books){

//complete this constructor

libraryName=name;

list=books;

}

//Write the instance method bestSeller, including its //method header, which returns the name of the bestselling //book. Return the first of the best selling books if //there are more than one.

public String bestSeller(){

Book best=Book.bestSeller(list);

return best.getName();

}

//Write the instance method doublePrice, which double the //price of each book in list.

public void doublePrice(){

for(int i=0;i<list.length;i++){

list[i].setPrice(list[i].getPrice()\*2);

}

}

}

3) Write the static method longestWord which accepts an array of Strings and returns the **length** of the longest word in the array.

public static int longestWord(String[] words){

int longest=words[0].length();

for(int i=1;i<words.length;i++){

if(words[i].length()>longest)

longest=words[i].length();

}

return longest;

}

}

4)Write the static method getSingleLetters which accepts a String str and returns an array of single-letter strings. Each of these strings consists of a single letter from str. Element k of the returned array contains the single letter at position k of str. For example, getSingleLetters(“cat”) returns the array {“c”, “a”, “t”}.

public static String[] getSingleLetters(String str){

String[] temp=new String[str.length()];

for(int i=0;i<str.length();i++){

temp[i]=str.substring(i,i+1);

}

return temp;

}