\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Core Java Exercise \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Consider the following class definition.

public class Ex11\_6

{

public int a,b;

public int c = 2;

public static int x = 6;

}

What are the values of a, b, c, and x, of instanceA and

instanceB after the following program is run?

public class Ex11\_6Test

{

public static void main(String[] args)

{

Ex11\_6 instanceA = new Ex11\_6();

Ex11\_6 instanceB = new Ex11\_6();

instanceA.a = 8;

instanceB.b = instanceA.x;

instanceA.x++;

instanceB.a = 10;

instanceB.c = 90;

instanceB.x++;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 . A file named Capsule.java has the following content.

public class Capsule

{

public static int nCapsules = 0;

public double volume;

public String screenText;

public Capsule(double volume,String s){

this.volume = volume;

screenText = s;

nCapsules++;

}

}

Determine the value of nCapsules when the following program

is run.

public class Ex11\_7Test

{

public static void main(String[] args)

{ int [] nInPack = {5,10,10};

Capsule [][] pack = new Capsule[3][];

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

pack[i] = new Capsule[nInPack[i]];

for(int j=0;j<pack[i].length;j++){

pack[i][j] = new Capsule(0.5,"Formular"+i+j);

}

}

System.out.println(Capsule.nCapsules);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3. Determine the output of the following program.

public class Ex11\_8Test

{

public static void main(String[] args)

{ int k = 5, j = 6;

Ex11\_8 a = new Ex11\_8();

Ex11\_8 b = new Ex11\_8(k\*j);

Ex11\_8 c = new Ex11\_8(k,j);

}

}

Given that the class Ex11\_8 is defined as:

public class Ex11\_8

{

public Ex11\_8(){

this(0);

System.out.println("A");

}

public Ex11\_8(int k){

this(0,0);

System.out.println("B");

}

public Ex11\_8(int k,int m){

System.out.println("C");

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4. Modify the class MyIntArrayUtil so that it contains another

static method that takes an input array of int and sort it in

increasing order. Then write a Java program that performs the

following steps.

a. Create an array of ten int values, where each value is

randomly chosen from 1 to 100.

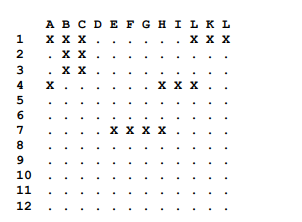
b. Show the elements of the array on screen.

c. Sort the elements of the array increasingly.

d. Show the elements of the array on screen again.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5. In a seat reservation program of an airline, the seating chart of an airplane is represented using a two-dimensional array, seats. The array seats[i][j] contains the name of the passenger who has reserved the seat number j in the i+1th row, where j is 0 for the seat number A, j is 1 for the seat number B, and so on. seats[i][j] stores null if the seat is vacant. Note that different airplanes may have different numbers of rows. However, assume that the number of seats in each row on the same airplane is constant. Write methods, that have one of their input arguments being the array seats, for performing the tasks in the following items. Decide on the names, their input arguments, and their returned values appropriately. a. Showing seating chart, labeled with row and seat numbers, by using ‘.’ to represent a vacant seat and ‘X’ to represent a reserved seat. An example of the seating chart could by like the one shown below.



If the caller of the method does not wish to see seating

of every rows, the caller can indicate the range of the

rows wished to be shown by specify the row numbers

of the first row and the last row to be shown.

b. Adding a passenger name to a selected seat. It returns

true if the operation is successful, and return false if

the selected seat is not empty. Given that the selected

seat is specified in the form of a String in the form:

“[row number]-[seat number]”, such as “1-A”, “25-E”,

and “36-I”. The method also checks whether the

specified seat is in the valid range. Unless the specified

seat is in the range of the array seats, the method does

nothing and returns false.

c. Removing the passenger at a specified seat.

d. Searching for the seat reserved by a passenger by

his/her name. The method returns the String

representing the seat location or null if there is no

passenger of the given name. Assume that each

passenger can occupy only one seat at a time.

e. Counting the number of seats available in each row.

The method returns an array of int where the value at

the ith index contains the number of available seats in

the i+1th row.

f. Searching for available n consecutive seats in the same

row. The method returns the String representing the

left-most seat location of the available n consecutive

seats in the front-most row that has such availability.

If there is no such availability the method returns null.

g. Randomly relocating passengers in the seating chart.

Each passenger must be assigned a seat not conflicting

with other passengers. The method returns the

randomized seating as its output, while the input

seating stays intact. (This method is not going to be

useful for any functioning airlines!)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6. Write a recursive Java method that calculates the sum of every

int element in an input array of int.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7. Write a recursive Java method that finds the smallest value in an

input array of int.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8. Write a recursive Java method that returns the index of the

smallest value in an input array of int.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*