|  |  |
| --- | --- |
| universal cover for all hwks and exams/V2 |  |
| Last name, First name |  |
| course (CSCI-NNN) | cs220 |
| Assignment number (hwkN/examN) | hwk2 |
| Date |  |
| your email address | @eagles.bridgewater.edu |
|  |  |

Do not do any computation on this page.

If the hwk has Multiple choice questions (MCQ), then mark your solution in this table here.

Suppose you think correct answer is C. You will then attach letter X to your desired answer like CX. If the work is handwritten you can handwrite X, if the work is typed in Word you will type in Word X.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | A | B | C | D |  |
| 2 | A | B | C | D |  |
| 3 | A | B | C | D |  |
| 4 | A | B | C | D |  |
| 5 | A | B | C | D |  |
| 6 | A | B | C | D |  |
| 7 | A | B | C | D |  |
| 8 | A | B | C | D |  |
| 9 | A | B | C | D |  |
| 10 | A | B | C | D |  |
| 11 | A | B | C | D |  |
| 12 | A | B | C | D |  |
| 13 | A | B | C | D |  |
| 14 | A | B | C | D |  |
| 15 | A | B | C | D |  |

//your typed or handwritten work starts on the next page

Note if I request that your answer should be 4-6 sentences, it is only a guideline. It means if you give me 2 sentences, it may not be enough. If you write half a page it is way too much.

hwk1

Ask questions if something is not clear.

Program1.

Write a Java program .

Class name MyArrayMax.

Package: no package.

The program will be written using OOP. Ask me in the class if you not sure what it means.

Do the best you can.

Methods. No methods (except main) will be static unless you can write absolutely convincing argument in the class header that the method must be static.

Hint: In my courses, less than 5% of methods will be static.

/\*\*

\*

\* author rrs

\*

\* Description:

\* Write class MyArrayMax.

\* This class uses plain array as opposed to something else.

\*

\* To get you started, I started the methods that you must have.

\* You can add additional assist methods you like,

\* but the methods I list are required.

\* You can add additional vars you want.

\*

\*

\* created: 2020/02/05/ok

\* works: 2020/02/05/ok

\*

\*/

**public** **class** MyArrayMax

{

//data

**private** **int**[] data;

/\*\*

\* Description:

\* This method will take argument an initialized array.

\* It will copy the array to the the array data in the class.

\*/

**public** **void** init(**int**[] datax)

{

}

/\*\*

\* Description:

\* This method will print the content of the array on a single line seperated by tabs.

\*/

**public** **void** print()

{

}

/\*\*

\* Description:

\* This method will find the largest odd element.

\*/

**public** **int** findMaxOdd()

{

}

/\*\*

\* Description:

\* This method find the sum of all elements with even value

\*/

**public** **int** findSumEven()

{

}

/\*\*

\* Description:

\* This method will reverse the values in the data array.

\*/

**public** **void** reverseArray()

{

}

//main

**static** **public** **void** main(String[] args)

{

//You will submit your program with the data I have here.

//I suggest you test your program with more than what I show over here.

// When done testing, your code in MAIN will \*\*\*look like this\*\*\*. this.\*\*\*

**int**[] test1 = { 2, 5, 7, -20, -30, 20 };

MyArrayMax myarr = **new** MyArrayMax();

myarr.init(test1);

System.out.println("original array");

myarr.print();

System.out.println("maxOdd=" + myarr.findMaxOdd());

System.out.println("sumEven=" + myarr.findSumEven());

System.out.println("reversed array");

myarr.reverseArray();

myarr.print();

}

}

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Program 2

Write a Java program .

Class name ArrayOfNames.

Package: no package.

The program will be written using OOP.

The key objective is that your program can not crash on ANY sequence.

Do the best you can.

Methods. No methods (except main) will be static unless you can write absolutely convincing argument in the class header that the method must be static.

Hint: In my courses, less than 5% of methods will be static.

/\*\*

\*

\* author rrs

\*

\* Description:

\* Write class ArrayOfNames.

\* This class uses plain array as opposed to something else.

\*

\*

\* To get you started, I started the methods that you must have.

\* You can add additional methods you like,

\* but the methods I list are required.

\* You can add additional vars.

\*

\*

\* created: 2020/02/05

\* works: 2020/02/05

\*

\*/

**public** **class** ArrayOfNames

{

// data

**private** String[] names;

**private** **int** len;

**public** ArrayOfNames(**int** szx)

{

}

/\*\*

\* Description:

\* This method will try to append name into the array.

\* Append means it will insert name in first empty slot

\*

\* If the whole array is full it will print: "append error: array is full".

\*/

**public** **void** append(String rec)

{

}

/\*\*

\* Description:

\* This method will try to insert new name into specific slot.

\* If slot is empty, insert new String.

\* If slot is not empty, print "insert: error" and do nothing.

\*/

**public** **void** insert(**int** ix, String rec)

{

}

/\*\*

\* Description:

\* This method will try to delete name.

\* If slot is empty, print "delete: error"..

\* If slot is not empty, delete the String and print it.

\*/

**public** **void** delete(**int** ix)

{

}

/\*\*

\* Description:

\* This method will print all names, one string per line.

\*/

**public** **void** print()

{

System.out.println("start print array");

System.out.println("end print array\n");

}

/\*\*

\* Description:

\* Delete all names.

\*/

**public** **void** delete()

{

}

// main

**static** **public** **void** main(String[] args)

{

// you will submit your program with whatever sequence I have here

// I suggest you test your data with more than what I show over here.

// When done testing, your code in MAIN will \*\*\*look like this\*\*\*.

ArrayOfNames aor = **new** ArrayOfNames(10);

String r1 = "joe";

String r2 = "bill";

String r3 = "evelyn";

String r4 = "lori";

String r5 = "beth";

String r6 = "carl";

String r7 = "jess";

String r8 = "faye";

aor.insert(3, r1);

aor.print();

aor.append(r2);

aor.print();

aor.insert(2, r3);

aor.print();

aor.delete(2);

aor.print();

aor.insert(5, r5);

aor.print();

aor.append(r6);

aor.print();

aor.insert(1, r7);

aor.print();

aor.delete();

aor.append(r8);

aor.print();

}

}