CS 113 Midterm 2M - Fall 2023- Requires Respondus LockDown Browser

Due Nov 13 at 6pm **Points** 100 **Questions** 13
Available Nov 13 at 4pm - Nov 13 at 6pm 2 hours **Time Limit** 105 Minutes
Requires Respondus LockDown Browser

Instructions

CS 113 MIDTERM EXAM 2

FALL 2023

PLEASE READ THESE TEST INSTRUCTIONS AND RULES CAREFULLY!!!
There are 12 questions on this test. The value of each question is:

1-10 multiple choice (total 40 pts)
11-12 coding problem (total 60 pts)

You may get partial credit for questions 11-12. If you finish early, use the extra time to double check your work.

- You may not use notes, books. One empty sheet of scrap paper is allowed for notes taking. Exam is administered using Lockdown Browser to monitor the test.
- You may not leave the room before you turn in your exam.
- All cell phones and other smart devices must be turned off and kept away during the exam.
- No headphones of any kind may be used during the test.
- A reference sheet will be provided proceeding the coding problems.

By taking the test you acknowledge that you have read an understood the rules above.

On my honor, I pledge that I have not violated the provisions of the NJIT ACADEMIC Honor Code.

Good luck!

This quiz was locked Nov 13 at 6pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	95 minutes	41.6 out of 100

Correct answers are hidden.

Score for this quiz: **41.6** out of 100
Submitted Nov 13 at 5:35pm
This attempt took 95 minutes.

Incorrect

Question 1

0 / 4 pts

Given the following java statement:

```
float[][] arr = new float[2][4];
```

Which statement declares and initializes variable ***n*** to the index of the last column (second dimension) of the array?

- int n = arr[3].length - 1;
- None of the given options
- int n = arr[0].length - 1;
- int n = arr.length - 1;
- int n = arr[0].length;

Question 2

4 / 4 pts

What is the value of ***let*** after this code executes?

```
int val=5;
char let='A';
switch (val % 4){
```

```
        case 1: let+=1;
        case 2: let+=2;
        case 3: let-=3;break;
        case 0: let+=0;
        default: let -= val%2;
}
```

- None of the given options
- A
- B
- C
- D

Question 3

4 / 4 pts

To make a class member shared among all instances of the class, use the keyword _____

- new
- public
- private
- static
- None of the given options

Incorrect

Question 4

0 / 4 pts

What is the content of list after the following code executes?

```
int[] list = {1, 2, 3, 4, 5};
for (int i = 1; i <= list.length - 1; i++)
    list[i-1] = list[i];
```

- 1,2,2,2,2
- 2,3,4,5,5
- 1,1,2,3,4
- None of the given options
- Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1

Incorrect

Question 5

0 / 4 pts

Given the Test class defined below:

```
class Test{
    private int score;

    public Test(int score){ this.score=score;}
    public int getScore(){ return score;}
    public void setScore(int newScore) { score =
newScore;}
}
```

Referring to the class above, what is the output of the code fragment below?

```
public static void main(String[] args){
    int newScore = 5;
    Test myTest = new Test(2);
    runTest(myTest, newScore);
    System.out.println(myTest.getScore() +"," +
newScore);
}
public static void runTest(Test t, int newScore){
   newScore = t.getScore();
    t.setScore(newScore + 1);
}
```

- 5,5
- None of the given options
- 3,2
- 5,2
- 3,5

Incorrect

Question 6

0 / 4 pts

Given the Test class defined below:

```
class Test{
    private int score;

    public Test(int score){ this.score=score;}
    public int getScore(){ return score;}
    public void setScore(int newScore) { score =
newScore;}
}
```

Assuming ***list*** is an array of Test objects defined by the class Test above, what does the following code do?

```
int val = 0;
for (int i = 0; i < list.length; i++)
    if (list[i].getScore() < list[val].getScore())
        val = i;
```

- It finds the total number of tests that have score bigger than 0.
- None of the given options
- It finds the index of the first test with lower score than test at position 0.
- It finds the lowest test score in list.
- It finds the index of the test with the lowest score in list.

Question 7

4 / 4 pts

Code below purports a recursive method. What is the problem with it?

```
public int question7(int n){
    if (n <= 0)
      return 0;
    else
      return question7(n) + 2;
}
```

- None of the given options

- There is no recursive call.
- There is no base case.
- The recursive call does not move the parameter closer to the base case.
- The recursive call moves the problem further away from the base case.

Incorrect

Question 8

0 / 4 pts

Which regular loop corresponds to this for-each loop?:

```
char[] vowels = {'a', 'e', 'i', 'o', 'u'};
for (char item: vowels) {
    System.out.println(item);
```

- for (int i = 0; i < item.length; ++i)
  System.out.println(vowels[i]);
- for (int i = 0; i < vowels.length; ++i)
  System.out.println(item[i]);
- for (int i = 0; i <= vowels.length; ++i)
  System.out.println(vowels[i]);
- None of the given options
- for (int i = 0; i < vowels.length; ++i)
  System.out.println(vowels[i]);

Question 9

4 / 4 pts

If the method is invoked as `foo(7)`, what is returned?

```
public static int foo(int n){
    if (n < 4)
        return n;
    else
        return foo(n-2) + n%3;
}
```

- 6
- This is an infinite recursion.
- 9
- 7
- None of the given options

Question 10

4 / 4 pts

An object that refers to part of itself within its own methods can use which of the following reserved words to denote this relationship?

- private
- inner
- None of the given options
- this

static

Partial

Question 11

18 / 30 pts

The class **Homework** below defines a homework assignment in terms of its score, e.g. homework score 95. The homework score is a an integer 0-100, inclusive.

```
public class Homework {
    private int score;

    public Homework(int score) {
        this.score = score;
    }
    public boolean equals(Homework other){
        return this.score == other.score;
    }
    public void setScore(int score) { this.score = score;}

    public int getScore() { return score;}

    public String toString(){ return "HW: "+Integer.toString(score);}
}
```

Complete the class **StudentRecord,** below, that represents the student's current record in terms of two homework assignments. The StudentRecord class should have two attributes:

- **hw1**, a Homework object
- **hw2**, a Homework object

The class should also have these members:

- A **constructor** that takes two Homework object parameters to set up the two homework
- A getter for **hw1**
- A setter for **hw2**
- We compare two student records based on their corresponding homework scores. Write a method **equals()** that compares two student records and returns **true** if they have the same corresponding homework scores and **false**, otherwise.
- The teacher is offering an incentive to all students who improve in homework2 by at least 16 points. As part of the incentive, if homework2 score is at least 16 points more than homework1 score, it will cause homework1 score to be increased by half (as an integer) of that improvement. For example, if homework1 score is 51 and homework2 score is 80, the homework1 score will be increased by 14 points, hence homework1 score is now 65. Write a method **applyIncentive()** that will apply incentive to homework1, only if eligible.
- A **toString()** method that returns a string containing each homework description, eg: "HW: 56, HW: 80".

```
public class StudentRecord {
    private Homework hw1, hw2;

    public StudentRecord(Homework hw1, Homework hw2) {
        this.hw1 = hw1;
        this.hw2 = hw2;
    }
    public Homework getHw1() {
        return hw1;
    }
    public void setHw2(Homework hw2) {
        this.hw2 = hw2;
    }
    public boolean equal(Homework other){
```

```
        return hw1 equals (other.hw1) &&
hw2 equals (other.hw2);
    }
    public void applyIncentive(){
        int difference = hw2 -
hw1 ;
        if (difference >= 16)
            hw1 setScore (hw1.getScore() +
difference/2);
    }
    public String toString(){
        return hw1 +", "+ hw2
;
    }
}
```

Answer 1:

class

Answer 2:

Homework

Answer 3:

Homework

Answer 4:

Homework

Answer 5:

this.hw1

Answer 6:

Homework

Answer 7:

Homework

Answer 8:

Homework

Answer 9:

equals

Answer 10:

equals

Answer 11:

hw2

Answer 12:

hw1

Answer 13:

setScore

Answer 14:

hw1

Answer 15:

hw2

The class **Product** below defines an item sold during a fundraiser event.

```
public class Product {
    private int price;
```

```
    private String description;

    public Product(String description, int price) {
        this.price = price;
        this.description = description;
    }

    public String getDescription() {
        return description;
    }

    public int compareTo(Product other){
        if (price < other.price) return -1;
        else if (price == other.price) return 0;
        else return 1;
    }
}
```

Solve the two parts of this problem below:

Question 12

2 / 22 pts

Part 1. Write a method called **`shoppingCart()`** that takes one parameter:

1. **`arr`**, a non-empty array of **`Product`** objects representing a collection of products with distinct prices to be purchased.

The method returns the description of the most expensive product. For example, if the product descriptions and their corresponding prices below represent the collection of products to purchase:

```
hat 17
sweater 60
t-shirt 25
sweatpants 36
```

the method will return *`sweater`*

Your Answer:

```
String shoppingCart(Product [] shopping){
int numColumns = shopping[0].length;
int numRows = shopping.length;
int[]expensive = new int(numColumns);
for(i=0;i<numRows;i++)
int maxPrice  = shopping[i][0];
for(j=0;j<numColumns;j++);
if(price.compareTo(other)>1){
int maxprice = shopping[i][j];
maxprice.getDescription()
}
return maxprice;
}
```

should be public in method header and paramter called arr Not two dimensional array so should not be iterating twice Price and other do not exist for compareTo and compareTo is never greater than 0 Should return description of most expensive product not maxprice variable +2

Partial

Question 13

1.6 / 8 pts

Part 2. Using the example products, given in **Part 1,** complete the code below that creates and populates an array of **`Product`** objects

called **shopping**, using an initializer list syntax.

```
Product[] shopping = new
Product ("hat",34), ("tshirt",25)
("sweater",65),
                    ("sweatpants",36)
("gloves",23)};
```

Answer 1:

Product[]

Answer 2:

new

Answer 3:

Product

Answer 4:

("tshirt",25)

Answer 5:

("sweatpants",36)

Quick Reference

Scanner Class

Scanner(InputStream source)
Scanner(File source)
Scanner(String source)
String next()
String nextLine()
int nextInt()
double nextDouble()
float nextFloat()

String Class

String(String str)
int length()
int compareTo(anotherString)
char charAt(int index)
boolean equals(String anotherString)
String substring(int beginIndex, int endIndex)
String substring(int beginIndex)

Random Class

Random()
float nextFloat()
int nextInt(int num)
int nextInt()

Math Class

static double random()
static final double PI

Quiz Score: **41.6** out of 100