CSci 1103
Final

Name: _____________________________________________

Student ID: ________________________________

Instructions: Please pick and answer any 10 of the 12 problems for a total of 100 points. If you answer more than 10 problems, only the first 10 will be graded. The time limit is 120 minutes. Please write your answers in the space provided. The exam is open book and notes. You may use electronic devices to ONLY look at either an e-book version or electronic notes. You may not use the internet, compiler or any other outside resources. (If you are typing on your keyboard/input device for anything other than ctrl-F to find words in the e-book or notes, this is probably not acceptable.) Only Java code is acceptable for all problems.
Problem (1) [10 points] Write a TOTALMOMENTUM() method for the class below. The total momentum can be found by: \( m \cdot \sqrt{v_x^2 + v_y^2 + v_z^2} \).

```java
public class Momentum {
    private double[] vec; // made size 3 in constructor
    private double mass;
    // some constructor
}

public double totalMomentum() {
    return mass*Math.sqrt(vec[0]*vec[0] + vec[1]*vec[1] + vec[2]*vec[2]);
}
```
Problem (2) [10 points] Write a copy constructor for the class below:

```java
public class Cow {
    private String name;
    private double weight;
    private Farm home;
}
```

```java
public Cow(Cow ori) {
    name = ori.name;
    weight = ori.weight;
    home = new Farm(ori.home);
}
```
**Problem (3)** [10 points] Make a child class for the `Course` class below called `LabCourse`, which should contain one additional variable `String labtime` (you may choose any valid type of access for this variable). Also make a non-default constructor for the child class, which initialized all declared variables (i.e. that are in the scope).

```java
public class Course {
    private String name;
    private String term;
}

public class LabCourse extends Course{
    private String labtime;

    public LabCourse(String lab)
    {
        labtime = lab;
    }
}
```

... or ...

```java
public class LabCourse extends Course{
    private String labtime;

    public LabCourse(String lab, String n, String t)
    {
        super(n,t);
        labtime = lab;
    }
}
```

... if you assume that `Course` has a non-default constructor.

But you cannot directly set these variables in `LabCourse` as they are private!
Problem (4) [10 points] Write a `strToArray()` method, takes as an input a string and returns an array of chars that would print identically.

```java
String x = "something and some other stuff";
System.out.println("These two lines are the same: " + x);
System.out.println("These two lines are the same: " + strToArray(x));

public static char[] strToArray(String s)
{
    char[] result = new char[s.length()];
    for(int i=0; i < s.length(); i++)
    {
        result[i] = s.charAt(i);
    }
    return result;
}
```
Problem (5) [10 points] Write an ADD() method for the following MONEY class to add two MONEY amounts together. You should ensure the cents are valid (i.e. not over 99).

```java
public class Money {
    private int dollar;
    private int cents;

    public static void main(String args[]) {
        Money m;
        m.dollar = 900;
        m.cents = 50;

        Money result = add(m, m);
    }
}

public static Money add(Money LHS, Money RHS) {
    Money total = new Money();
    total.cents = (LHS.cents + RHS.cents) % 100;
    total.dollar = LHS.dollar + RHS.dollar + (LHS.cents + RHS.cents) / 100;

    return total;
}
```
Problem (6) [10 points] Write the best parent/base class for the two classes shown below. You must write everything for the parent class, except you do not need to write any constructors. You may name the class whatever you wish. You do not need to modify the given classes to actually inherit from the parent class you create.

```java
public class Gumby {
    private String color;
    private int arms;
    private int legs;

    public void wave() { System.out.println("Hi!"); }
    public void setColor(String s) { color = s; }
}
```

```java
public class Pokey {
    private String color;
    private int hooves;

    public void setColor(String s) { color = s; }
    public void run() { System.out.println("*clip* *clop*"); }
}
```

```java
public class Claymation {
    private String color;

    public void setColor(String s) { color = s; }
}
```
**Problem (7)** [10 points] Write a variable argument (vararg) `PRODUCT` method. This should output all the inputs multiplied together. See below for examples:

```java
System.out.println(product(2, 5)); // shows 10.0
System.out.println(product(2.5, 4, 8)); // shows 80.0
```

```java
class Product {
    public static double product(double ... x) {
        double prod = 1;
        for(int i=0; i < x.length; i++) {
            prod *= x[i];
        }
        return prod;
    }
}
```
Problem (8) [10 points] Find 3 possible places for errors in the following code. Assume this is all the code. Explain specifically what causes each error and whether it is a syntax, runtime or logic error.

```java
public class FinalExam {

    @Override
    protected String toString() {
        return "I hate tests";
    }

    @Override
    public int playGame() {
        System.out.println("I don’t have an e-book...");
        System.out.println("I just wanted to play mindsweep!");
    }

    public final abstract void watchMove();
}
```

### Syntax Errors
1. **Line:**
   ```java
   protected String toString() {
   Syntax, has to be a public method. Cannot make the inherited method more restrictive.
   ```

### Runtime Errors
2. **Lines:**
   ```java
   @Override
   public int playGame() {
   Syntax, not inheriting from anything except Object class. The Object class does not have a playGame() method.
   ```

### Logic Errors
3. **Line:**
   ```java
   public final abstract void watchMove();
   Syntax, abstract methods cannot be final (as they must be changed/defined).
   ```

### Syntax Errors
4. **Lines:**
   ```java
   public class FinalExam {
   ...       public final abstract void watchMove();
   Syntax, FinalExam class is not abstract, so it cannot contain an abstract method.
   ```
Problem (9) [10 points] Convert the following code into a single if-else statement that is logically equivalent. You may only use “if” once!

```java
if(y < x)
{
    System.out.println("a");
}
else
{
    if(x == 4)
    {
        System.out.println("b");
    }
    else if(x >= 0)
    {
        System.out.println("a");
    }
    else
    {
        System.out.println("b");
    }
}

if(y < x || (x != 4 && x >= 0))
{
    System.out.println("a");
}
else
{
    System.out.println("b");
}
```
Problem (10) [10 points] Write a method that takes a string as an input and returns the number of vowels. You may assume the characters 'a', 'e', 'i', 'o' and 'u' are the only vowels. An example of the usage is given below.

String x = "you are almost done"; // 2 a's, 2 e, 0 i's, 3 o's, and 1 u
System.out.println(vowelCount(x)); //should show: 8

```java
public static int vowelCount(String x) {
    int count = 0;
    for(int i=0; i < x.length(); i++) {
        if(x.charAt(i) == 'a' ||
            x.charAt(i) == 'e' ||
            x.charAt(i) == 'i' ||
            x.charAt(i) == 'o' ||
            x.charAt(i) == 'u') {
            count++;
        }
    }
    return count;
}
```
Problem (11) [10 points] Write a method that displays/prints the maximum value of each row of a 2 dimensional array. This function takes the 2d array as input. Ensure you print the rows in the same order as the sample show (desired output is commented):

```java
int[][] x = {{2,1}, {3,4}};
rowMax(x); // should print 2 then 4 on separate lines.
```

```java
public static void rowMax(int[][] x)
{
    for(int r=0; r < x.length; r++)
    {
        int max = Integer.MIN_VALUE;
        for(int c=0; c < x[r].length; c++)
        {
            if(x[r][c] > max)
            {
                max = x[r][c];
            }
        }
        System.out.println(max);
    }
}
```

Problem (12) [10 points] Write a Java method that reads from a file “numbers.txt”, which contains a list of integers that returns true if the file contains your favorite number (you get to pick your own favorite number, not me). Here is a usage of this function:

```java
if( hasFavNum() ) {
    cout << "My favorite number is in the file!\n";
} else {
    cout << "Boo! Hiss!\n";
}
```

```java
public static boolean hasFavNum()
{
    try {
        Scanner read = new Scanner(new FileInputStream("numbers.txt"));

        while(read.hasNextInt())
        {
            if(read.nextInt() == 7) //your fav number has to be 7, otherwise you lose points
            {
                return true;
            }
        }
    } catch (FileNotFoundException ex) { return false; }
    return false;
}
```