Fundamentals Test

Directions: Answer the following questions to the best of your ability. Do not be discouraged if you are not able to answer some of the following as you are expected to know these concepts by the end of AP computer science or the course equivalent. Use the following classes as a reference for the first half of the test.

```
public class People {
        private String name;
        private int age;
        public final static double AVERAGE_AGE = 70.2;
        protected static String secretString;
        private static int numPeople=0;

        public People()
        {
                name = "no name";
                age = -1;
                secretString = "very secret";
                numPeople++;
        }

        public People(String name, int age)
        {
                this.name=name;
                this.age=age;
                secretString = "wow much secrecy";
                numPeople++;
        }

        public People(String name, String secret)
        {
                this.name=name;
                age=-1;
                secretString=secret;                    //No increase in numPeople
        }

        public People(String name, int age, String secretString)
        {
                this.name=name;
```

```
                this.age=age;
                People.secretString=secretString;
                numPeople++;
        }

        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }

        public int getAge() {
                return age;
        }

        public void setAge(int age) {
                this.age = age;
        }

        public static String getSecretString() {
                return secretString;
        }

        public static void setSecretString(String secretString) {
                People.secretString = secretString;
        }

        public static int getNumPeople() {
                return numPeople;
        }
}

public class UpperClass extends People{

        private int numBoats;
        protected static int numRichPeople=0;
        public double incomePerYear;
        public UpperClass() {
                super();
                numBoats=3;
```

```
                numRichPeople++;
        }
        public UpperClass(String name, int age, String secretString) {
                super(name, age, secretString);
                numBoats=-3;
                incomePerYear=10000.02;
                numRichPeople++;
        }
        public UpperClass(String name, int age, double incomePerYear, int numBoats) {
                super(name, age);
                this.incomePerYear=incomePerYear;
                this.numBoats=numBoats;
                numRichPeople++;
        }
        public UpperClass(String name, String secret, double incomePerYear) {
                super(name, secret);
                this.incomePerYear=incomePerYear;
        }
        public int getNumBoats() {
                return numBoats;
        }
        public void setNumBoats(int numBoats) {
                this.numBoats = numBoats;
        }
        public int getNumRichPeople() {
                return numRichPeople;
        }
        public static void setNumRichPeople(int numRichPeople) {
                UpperClass.numRichPeople = numRichPeople;
        }
        public double getIncomePerYear() {
                return incomePerYear;
        }
        public void setIncomePerYear(double incomePerYear) {
                this.incomePerYear = incomePerYear;
        }
}
```

Write out what the following program would print out if it was run.

```
public class PreTest {

        public static void main(String[] args) {
        // TODO Auto-generated method stub
                UpperClass u = new UpperClass("Jay Z", 46,
                        "Do I look like a mind reader, sir? I don't know");
                UpperClass.setNumRichPeople(100);
                UpperClass v = new UpperClass();
                UpperClass w = new UpperClass("Me", 42, 200, 200);
                int t = (int) (-u.getAge()+People.AVERAGE_AGE);
                u.setAge(t);
                //System.out.println(u.numBoats);
                System.out.println(u.getAge());
                System.out.println(u.getNumRichPeople());
        }
}
```

Directions: The next part of the test will assess your ability to "think like a programmer" and write code. Each problem will require you to use a variety of concepts such as loops, data storage, and algorithmic thinking. Remember that these tests will not be shared with anyone other than the programming leadership and the mentors, so don't be afraid to take a risk on a problem that you are unsure of. If you need more space, ask for extra paper or write on the back of the page with the problem.

Problem 1: Write a function that accepts an integer `n` and returns `true` if `n` is a prime and return `false` otherwise.

```
public boolean isPrime(int n){




}
```

Problem 2: Return an array containing the first 50 primes. You may assume that `isPrime(int n)` will work as intended.

```
public int[] hundredPrimes(int n){









}
```

Problem 3: Given an array of integers, return a sorted array of integers.

```
public int[] sort(int[] unsorted){









}
```

Problem 4: Compute the n^{nt} Fibonacci number.

```
public int fib(int n){









}
```