



Name: _____
Period: _____

AP Java Quiz 03: REVIEW (ICT L05, L06, L07, L08)



INTRODUCTION:

- On the following pages are **MechanicalPencilTester.java** and **MechanicalPencil.java**. **MechanicalPencilTester.java** is complete, and DOES NOT need any changes.
- **MechanicalPencil.java** has some complete code, and other code that is incomplete.
- **MechanicalPencil.java** DOES NOT need any additional private global variables.
- **MechanicalPencil.java** methods you write can have local variables added if needed.

DIRECTIONS:

Please complete all methods and parameters in **MechanicalPencil.java** that lack code and are marked to be completed. Use the comments ABOVE each method for directions and hints.

```

public class MechanicalPencilTester
{
    public static void main(String args[])
    {
        //CREATE MECHANICAL PENCIL OBJECTS/
        MechanicalPencil Pentel      = new MechanicalPencil();
        MechanicalPencil Papermate   = new MechanicalPencil(0.5, 9, false);
        MechanicalPencil Bic        = new MechanicalPencil(Papermate);

        //PRINT MECHANICAL PENCIL BEFORE/
        System.out.println("Pentel BEFORE:");
        Pentel.printMechanicalPencilInfo();
        System.out.println("Papermate BEFORE:");
        Papermate.printMechanicalPencilInfo();
        System.out.println("Bic BEFORE:");
        Bic.printMechanicalPencilInfo();

        //CHANGES / OTHER METHODS/
        System.out.println("Pentel is considered to be a " + Pentel.leadType() + " pencil.");
        System.out.println("Papermate is considered to be a " + Papermate.leadType() + " pencil.");
        System.out.println("Bic is considered to be a " + Bic.leadType() + " pencil.");
        System.out.println();

        System.out.println("Pentel:");
        Pentel.leadCountToPackages();
        System.out.println("Papermate:");
        Papermate.leadCountToPackages();
        System.out.println("Bic:");
        Bic.leadCountToPackages();
        System.out.println();

        System.out.println("Pentel.decreaseLeadCount(4);");
        Pentel.decreaseLeadCount(4);
        System.out.println("Papermate.increaseLeadCount(1);");
        Papermate.increaseLeadCount(1);
        System.out.println("Bic.increaseLeadCount(10);");
        Bic.increaseLeadCount(10);
        System.out.println();

        System.out.println("Pentel.subtractEraser();");
        Pentel.subtractEraser();
        System.out.println("Papermate.addEraser();");
        Papermate.addEraser();
        System.out.println("Bic.addEraser();");
        Bic.addEraser();
        System.out.println();

        //PRINT MECHANICAL PENCIL AFTER/
        System.out.println("Pentel AFTER:");
        Pentel.printMechanicalPencilInfo();
        System.out.println("Papermate AFTER:");
        Papermate.printMechanicalPencilInfo();
        System.out.println("Bic AFTER:");
        Bic.printMechanicalPencilInfo();
    }
}

```

decreaseLeadCount(int leads)
 increaseLeadCount(int leads)
 subtractEraser()
 if (canErase = True)
 {
 canErase = False;
 }
 addEraser()
 if (canErase = False)
 {
 canErase = True;
 }

```
import java.util.Scanner;

public class MechanicalPencil
{
    //These are the only private global values allowed, DO NOT add more
    private double myLeadSize;          // 0.2mm to 0.9mm range is allowed
    private int myLeadCount;           // The number of leads in the pencil
    private boolean canErase;          // Does this pencil have a usable eraser?

    public MechanicalPencil()
    {
        getValues();
    }
}
```

//COMPLETE THE MISSING PARAMETER(S) FOR THIS CONSTRUCTOR//

```
public MechanicalPencil(double leadSize, int leadCount, boolean eraser)
{
    myLeadSize = leadSize;
    myLeadCount = leadCount;
    canErase = eraser;
}
```

//COMPLETE THE MISSING PARAMETER(S) FOR THIS CONSTRUCTOR//

```
public MechanicalPencil(MechanicalPencil otherPencil)
{
    myLeadSize = OtherPencil.getMyLeadSize();
    myLeadCount = OtherPencil.getMyLeadCount();
    canErase = OtherPencil.getCanErase();
}
```

//COMPLETE THIS METHOD//

//Write a method which **EFFICIENTLY** creates a Scanner Object, and then gets
//the private values myLeadSize, myLeadCount, and canErase from the user.
//Scanner class methods like .nextInt(), .nextDouble(), and .nextBoolean() are useful.
//Don't forget to clearly PROMPT the user so they know EXACTLY what they should enter!
private void getValues()
{

Scanner MYINPUT = new Scanner(System.in)

System.out.print("The size of the lead is: ");

myLeadSize = MYINPUT.nextDouble();

System.out.print("The lead count is: ");

myLeadCount = MYINPUT.nextInt();

System.out.print("It can erase: ");

canErase = MYINPUT.nextBoolean();

System.out.println();

//COMPLETE THIS METHOD//

```
//A pencil's size can be described as "Small", "Medium", or "Large" to a user.  
//Write a method which **EFFICIENTLY** returns a String depending on the lead size range.  
//This will require if statement(s) and might require use of "and" (&&) for the condition.  
//Return "small" if the lead is in the exact range 0.2mm to 0.4mm inclusive.  
//Return "medium" if the lead is in the exact range 0.45mm to 0.65mm inclusive.  
//Return "large" if the lead is in the exact range 0.7mm to 0.9mm inclusive.  
//Return "incorrect size" if the lead is NOT in one of these three size ranges!
```

```
public String leadType()
```

```
{
```

```
String Result;
```

```
if(myLeadSize  $\geq$  0.2 && myLeadSize  $\leq$  0.4)
```

```
    Result = "small";
```

```
else if(myLeadSize  $\geq$  0.45 && myLeadSize  $\leq$  0.65)
```

```
    Result = "medium";
```

```
else if(myLeadSize  $\geq$  0.7 && myLeadSize  $\leq$  0.9)
```

```
    Result = "large";
```

```
else Result = "incorrect size";
```

```
return Result;
```

```
return Result;
```

```
}
```

4 Lx 10/10

//COMPLETE THIS METHOD//

//Write a method which **EFFICIENTLY** tells the user if they can erase or not.
//If they can erase, the exact message, "Yes, this pencil can erase." is printed.
//If they cannot erase, the exact message, "No, this pencil cannot erase." is printed.
public void printCanErase()

```
{  
    if (canErase = true)  
    {  
        System.out.println("Yes, this pencil can erase.");  
    }  
    else  
    {  
        System.out.println("No, this pencil cannot erase.");  
    }  
}
```

//COMPLETE THIS METHOD//

//It would be nice to know how many lead refill packages our pencil can store inside it.
//Write a method which **EFFICIENTLY** finds how many "packages" of leads are in the pencil.
//The formula to convert from lead count to packages is: [packages = lead count / 90].
//The method should print the answer formatted to two decimal places using printf().
//Say something clear to the user like, "The number of lead packages inside the pencil is ".
//HINT: Your answer needs to be a double, so avoid doing INTEGER DIVISION!!
public void leadCountToPackages()

```
{  
    double packages;  
    packages = myLeadCount / 90.0;  
    System.out.printf("The number of lead packages inside the pencil is %.2f", packages);  
}
```

//OTHER METHODS WITHIN THE CLASS THAT WORK AS WRITTEN, DO NOT CHANGE//

```
public void printMechanicalPencilInfo()
{
    System.out.println("This MechanicalPencil has a " + myLeadSize + "mm lead.");
    System.out.println("This MechanicalPencil has " + myLeadCount + " leads inside it.");
    System.out.println("It is " + canErase + " that this Mechanical Pencil can erase.");
    System.out.println();
}

public void increaseLeadCount(int leads)
{
    myLeadCount += leads;
}

public void decreaseLeadCount(int leads)
{
    myLeadCount -= leads;
}

public void addEraser()
{
    //If the pencil doesn't already have an eraser, add an eraser
    if(canErase == false)
    {
        canErase = true;
    }
}

public void subtractEraser()
{
    //If the pencil has an eraser, remove the eraser
    if(canErase == true)
    {
        canErase = false;
    }
}

public double getMyLeadSize()
{
    return myLeadSize;
}

public int getMyLeadCount()
{
    return myLeadCount;
}

public boolean getCanErase()
{
    return canErase;
}
```

if(canErase == false)
{ canErase = true;
}

if (canErase == true)
{ canErase = false;

//COMPLETE THIS METHOD//

```
//Write a method which **EFFICIENTLY** tells the user if they can erase or not.  
//If they can erase, the exact message, "Yes, this pencil can erase." is printed.  
//If they cannot erase, the exact message, "No, this pencil cannot erase." is printed.  
public void printCanErase()  
{  
    if (canErase == true)  
    {  
        System.out.println("Yes");  
    } else  
    {  
        System.out.println("No");  
    }  
}
```

//COMPLETE THIS METHOD//

```
//It would be nice to know how many lead refill packages our pencil can store inside it.  
//Write a method which **EFFICIENTLY** finds how many "packages" of leads are in the pencil.  
//The formula to convert from lead count to packages is: [ packages = lead count / 90 ]. left-  
//The method should print the answer formatted to two decimal places using printf().  
//Say something clear to the user like, "The number of lead packages inside the pencil is ".  
//HINT: Your answer needs to be a double, so avoid doing INTEGER DIVISION!!  
public void leadCountToPackages()  
{  
    double packages;  
    packages = myLeadCount / 90.0;  
    System.out.printf("The number of lead packages inside the pencil is %.2f",  
    packages);  
}
```

//COMPLETE THIS METHOD//

```
//A pencil's size can be described as "Small", "Medium", or "Large" to a user.  
//Write a method which **EFFICIENTLY** returns a String depending on the lead size range.  
//This will require if statement(s) and might require use of "and" (&&) for the condition.  
//Return "small" if the lead is in the exact range 0.2mm to 0.4mm inclusive.  
//Return "medium" if the lead is in the exact range 0.45mm to 0.65mm inclusive.  
//Return "large" if the lead is in the exact range 0.7mm to 0.9mm inclusive.  
//Return "incorrect size" if the lead is NOT in one of these three size ranges!  
public String leadType()
```

```
String result;  
if(mLeadSize == 0.2 && mLeadSize <= 0.4)  
{  
    result = "small";  
}  
else if(mLeadSize >= 0.45 && mLeadSize <= 0.65)  
{  
    result = "medium";  
}  
else if(mLeadSize >= 0.7 && mLeadSize <= 0.9)  
{  
    result = "large";  
}  
else  
{  
    result = "incorrect size";  
}  
return result;
```