APCS DQ Name: Group:

1. Write two constructors for the `Clock` class.

```
public class Clock
{
     private int hours;
     private int minutes;
     private int seconds;

     /**
      * your code here
      */









     /**
      * other code not shown
      */









}
```

APCS DQ Name: Group:

1. The `isEven` method should return the boolean value `true` if the `value` is divisible by 2.

   For example, `isEven(5)` should return `false`, `isEven(50)` should return `true`, and `isEven(100)` should also return `true`.

   Complete the `isEven` method.

```
public boolean isEven(int value)
{
    /**
      *    write your code here:
      */









}
```

APCS DQ 2 Name: Group:

1. Write two constructors for the `Player` class, one with no parameters, and another with one parameter. For example, outside this class one could make a new instance with `new Player()` as well as `new Player("Marsha")`

```
public class Player
{
     private int score;
     private String name;


     /**
      * your code here
      */









     /**
      * other code not shown
      */



}
```

3. Write a method `minus` that will return the difference of two or three integers. For example

call	returns
`minus( 5,2 );`	3
`minus( 2,5 );`	-3
`minus( 1, -5, 2 );`	4
`minus( -1, -9, -3 );`	5

```
class Question3
{
```

```
   public static void main (String[] args)
   {
      int a = minus(5,2);
      int b = minus(1,-5,2);
      int c = minus(-1,-9,-3);
      System.out.println(a+"- expected 3");
      System.out.println(b+"- expected 4");
      System.out.println(c+"- expected 5");
   }
}
```

4. Write a method `doubler` that will return `String` that repeats the first three letters attaching it to the original word. **Precondition:** all input `Strings` will have a length of at least three. For example

call	returns
`doubler("Boo" );`	`BooBoo`
`doubler( "Mama" );`	`MamaMam`
`doubler( "Spoon" );`	`SpoonSpo`
`doubler( "dog" );`	`dogdog`

```
public class Question4
{
```

```
   public static void main (String[] args)
   {
      String a = doubler("Cute");
      String b = doubler("tooth");
      String c = doubler("out");
      System.out.println(a+"- expected \"CuteCut\"");
      System.out.println(b+"- expected \"toothtoo\"");
      System.out.println(c+"- expected \"outout\"");
   }
}
```

2. Consider the following Class which uses an instance of the `Lock` class. Note that the lock can only be opened if the password matches:

```
public class LockTester
{
   public static void main(String[] args)
    {
       Lock lock1 = new Lock("secret");
       padlock.close();
       lock1.open("iforget");
       Lock lock2 = new Lock();
       lock2.setPassword("cheese");
       lock2.close();
       lock2.open("cheese");
       System.out.println("lock1 is "+lock1+" [expected: closed]);
       System.out.println("lock2 is "+lock2+" [expected: open]);

    }
}
```

Below is an incomplete implementation of the `Lock` class. You need to make three constructors so that class `LockTester` works properly.

```
public class Lock
{
   private String password;
   private boolean isOpen;

   /*******************************
    *  put your constructors here
    *******************************/









    /*********************************
     *  here are the method declarations
     *********************************/
    public void close() {is Open=false;}
    public void open(String pw){if (pw.equals(password)) isOpen=true;}
    public void setPassword(String n){password=n;}
    public String toString(){if (isOpen) return "open"; return "closed";}
}
```