Develop and test a small program that uses standard collection classes

Your task here is to complete the attached File Extensions application.

The program is intended to store file extensions as a **key** and the associated program as the **value**. You will need to create an appropriate instance variable to store the file extension entries as a key pair value where the file extension is the key and the default program is the value. You should use an appropriate generic standard collection class with both the key and the value stored as string objects. You should then complete methods as detailed in the internal documentation (see: section "Starter code supplied").

Once you have completed the implementation, you should run the attached partial acceptance test to ensure that the program behaves as expected. You should also document any errors found and how you fixed the errors.

Please upload the following files: **your Java source code** (typically 2 files: one for the implementation and one for the testing), and **screenshots of the program at runtime** (in particular, the _output_ window ie the console).

Starter code supplied

```
public class ExtensionMap {

    private /* collection class name here */ ;
    private String message ;                          // message for the user

    // default constructor
    public ExtensionMap() {
       // TO DO: write 2 lines of appropriate code
    }

    // return the message
    public String getMessage() {
       // TO DO: return the current message (String)
    }

    // add an entry to the collection
    public int addEntry( String k, String v ){
       // TO DO: add a key-value pair to the collection
       // set the "message" String appropriately ie the message for the user
       // return the new size of the collection
    }

    // remove an entry from the collection
    public int removeEntry( String k ){
       // TO DO: check whether the key (k) exists in the collection
       // - if so, remove the corresponding key-value pair
       //   & set an appropriate message
       // - if the key does not exist, set another message
       //   ( to let the user know - if they call getMessage() )
       // return the new collection size
    }

    // return the current size of the collection
    public int getSize() {
       // TO DO: return the current size of the collection
    }

    // return the whole map as a String -> see javadoc!
    @Override
    public String toString(){
       // TO DO: call the map's toString() method and return the result
    }

    // return a map entry
    public String getEntry( String key ){
       // TO DO
       // check whether the key exists in the collection
       // - if so, return the corresponding value
       // - if the key does not exist, set an appropriate message
       //   ( to let the user know - if they call getMessage() )
       //   and return the message
    }
} // end of class ExtensionMap
```

Code for partial acceptance testing

```
public static void main(String[] args){

  ExtensionMap em = new ExtensionMap() ;

  System.out.println( em.toString() ) ;  // map empty
  System.out.println( "-------------------------------------------------");
  mapAddRemoveTest( em ) ;               // add/remove some k-v pairs
  System.out.println( "-------------------------------------------------");
  System.out.println( em.toString() ) ;  // test the getMap() method
  System.out.println( "-------------------------------------------------");
  mapKVexistsTest( em ) ;                // does the map contain defined keys?
  System.out.println( "-------------------------------------------------");

}

// add and remove k,v pairs from the map
// "user-friendly messages"
public static void mapAddRemoveTest( ExtensionMap e ){
    e.addEntry( ".txt", "Notepad++" ) ;
    System.out.println( e.getMessage() ) ;
    e.addEntry( ".html", "Mozilla Firefox" ) ;
    System.out.println( e.getMessage() ) ;
    e.addEntry( ".png", "Paint" ) ;
    System.out.println( e.getMessage() ) ;
    e.addEntry( ".pdf", "Adobe Acrobat Reader" ) ;
    System.out.println( e.getMessage() ) ;

    e.removeEntry( ".png" ) ;
    System.out.println( e.getMessage() ) ;
    e.removeEntry( ".cpp" ) ;
    System.out.println( e.getMessage() ) ;
}

// print the map's contents to the console
public static void mapPrintTest( ExtensionMap e ){
    System.out.println( e ); ;
}

// test whether certain keys exist in the map
public static void mapKVexistsTest( ExtensionMap e){
   String[] testKeys = { ".docx", ".html", ".txt", ".cpp" } ;

   for( String s : testKeys )
      System.out.println( s.concat( " -> ").concat( e.getEntry(s) ) );

}
```

Expected output

```
{}
-------------------------------------------------
key-value pair added to the map
key-value pair added to the map
key-value pair added to the map
key-value pair added to the map
key-value pair removed from the map
key not in the map -> nothing removed
-------------------------------------------------
{.html=Mozilla Firefox, .pdf=Adobe Acrobat Reader, .txt=Notepad++}
-------------------------------------------------
.docx -> key not found
.html -> Mozilla Firefox
.txt -> Notepad++
.cpp -> key not found
-------------------------------------------------
```