



eduresourceza.com

SPY GAMES

MI5 has a program to record the details of all the spies in the organisation. Each spy is based in a different city in Europe. MI5 needs to be able to access this information for top secret purposes. The details of each spy are stored in a text file called SpyData.txt. The file consists of three fields separated by the # character: first name, surname and the city they currently operating in.

SpyData.txt

```
Josephine#Baker#London#
Mata#Hari#Paris#
Alger#Hiss#Milan#
Belle#Boyd#London#
Nathan#Hale#Zurich#
Richard#Sorge#Moscow#
John#Andre#Paris#
Klaus#Fuchs#Moscow#
Giacomo#Casonova#Zurich#
Aldrich#Ames#London#
```

SECTION A**OBJECTS AND ARRAYS****82 marks**

Create a new Java project called **SpyProj** to do the following:

- 1.1 Create a GUI application called **MainUI** where the user can select any of the following menu options:

Objects

Display a sorted list of spies
Search for a Spy
View encoded list

Database

View all spies
View spies sorted
Search for a spy in a city

(4)

- 2.1 Create a **Spy** class with instance variables for first name, surname and city. Declare each field using an appropriate type and ensure the fields are not accessible outside the class. (3)

- 2.2 Code the following methods for the class:

2.2.1 A parameterised constructor. (2)

2.2.2 Accessor methods for each instance variable. (2)

2.2.3 Mutator methods for each instance variable. (2)

2.2.4 **toString** method – which creates and returns a string of all the fields in the class separated by tabs. (3)

- 2.3 Code a method in the **Spy** class to encode the first name and surname of each spy. Use the following encryption scheme:

- Convert the letters to capitals
- Extract all the consonants from the name and place at the beginning in order
- Extract all vowels from the name and place at the end in order
- Reverse the consonants and the vowels and join the two together

For example “Josephine” converted to capitals will be JOSEPHINE

Becomes: JSPHN (consonants) OEIE (vowels)

Reversed and joined: NHPSJEIEO

(15)

Marks will be awarded for the efficiency of your code.

- 3.1 Create a class called **SpyArray** to hold the information about all the spies stored in the text file called **SpyData.txt**. It has an instance array called **spyArr** to store objects of type **Spy** and a field called **size** to store the number of spies in the array. You may assume that there will never be more than 100 spys in the array **spyArr**. (4)
- 3.2 Code the following methods for the **SpyArray** class:
 - 3.2.1 Constructor that initialises the array called **spyArr** using the data in the text file called **SpyData.txt**. The constructor must read the data from the text file and instantiate each element in the array called **spyArr** using the constructor in the **Spy** class. The field called **size** must contain the number of elements instantiated in the array. (10)
 - 3.2.2 Code a method called **sort** to sort the array called **spyArr** alphabetically according to the surname field. Use a method to swop the elements in the array. (8)
 - 3.2.5 Code a typed method **search** that will find all the spies who are in a particular city. If there are no spies in the given city display a suitable message. If data is found, display the surnames of the spies and the name of the city. The results of the search must be returned as a string. (8)
 - 3.2.4 Code a typed method called **viewCoded** to view the spies' first names and surnames in the encoded form. Return the results as a string. (6)
 - 3.2.5 Code a method called **toString** that creates a string of all the spies using the Spys' **toString** method. Each spy should be on a new line. (4)
- 3.3 Instantiate an **allSpys** object of the **SpyArray** class in **MainUI** class to access the methods in the **SpyArray** class. (2)
4. **Menu Option - Display sorted list of spies**
Add code in the **MainUI** class/unit to call the **sort** method in the **SpyArray** class and display the sorted list of spies in an appropriate component on the screen. (2)
5. **Menu Option – Search for a spy**
Add code in the **MainUI** class/unit to call the **search** method in the **SpyArray** class to search for a spy given the surname of the spy and the city they are working in. Display the result on the screen. (4)
6. **Menu Option – View encoded list**
Add code in the **MainUI** class/unit to call the appropriate method in the **SpyArray** class to view all the spies' full names in their encoded form. (3)

MI5 has rated the spies level of competency from 1 to 5 with 5 being the most dangerous. This information is added as an extra field in a database.

2.1 Create a new database called **SpyDB** in mySql. (1)

2.2 In Netbeans using the Execute Command option, copy the code from the **SpySQL.sql** text file and run this script to create a table called **tblSpy** and populate that table with data. (2)

Check that your table has been successfully created.

3. **DB** class:

3.1 Add the file **DB.Java** to your project to create a connection to the **SpyDB** database. Make sure the file is in the same package as the **MainUI** class. (1)

3.2 Add a method called **query** to the DB class to query the database and return a result set object given the SQL statement. (5)

Use the query method in the DB class to achieve the following with SQL statements:

4. **Menu Option – View all spies**
Add code to the **MainUI** class to view all the spies stored in the table **tblSpies**. (3)

5. **Menu Option – View spies sorted**
Add code to the **MainUI** class to view all the spies in the table **tblSpies** sorted according to the city then the rating. (5)

6. **Menu Option – Search for a spy in a city**
Add code to the **MainUI** class to input the name of a city and the spy's first name or surname. Using this information, search for a spy in a particular city given either the first name or surname. (7)
