



eduresourceza.com

OBJECTS AND ARRAYS

80 marks

High Scores

A games programmer wants to write a section of code to keep track of the user's scores of a game. He has created a text file in Notepad with sample data. The file consists of the user's name, score and their time in minutes followed by seconds (mm:ss). This file is stored in your exam drive and is shown below.

HiScores.txt

```
Fred#613#11:54#
Darth Vader#423#2:46#
Simeone#805#14:00#
Neo#448#7:29#
RoxyRhythm#435#10:46#
Trixster#482#8:33#
Fred#588#9:55#
Neo#836#2:48#
Darth Vader#554#2:33#
Fred#147#15:47#
Trixster#124#4:13#
Neo#574#12:01#
Neo#807#6:09#
```

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

HiScore Class

- 1.1 Create a **HiScore** class with instance variables to store the user's name, score and the time that they took to play the game. Use appropriate types and ensure that the fields are hidden in the class. (4)
- 1.2 Code a parameterised constructor for the class. A **HiScore** object will be initialised at the beginning of the game and the user will not have a value for the score. Set the score field to 0. (5)
- 1.3 Code an accessor and mutator method for the **score** field. (4)
- 1.4 Code an appropriate method called **getSeconds** to return the time in seconds. For example if the user time is 4:13 then the time in seconds is $4 \times 60 + 13 = 253$ seconds. (5)
- 1.5 Code a **toString** method to return the name, score and time in seconds separated by tabs. (6)

HiScoreArray Class

- 2.1 Create a class called **HiScoreArray** to store all the HiScores in an array. Declare an array of 50 HiScore objects and use a field to remember the number of elements in the array. (5)
- 2.2 Code a constructor method that will read the data from the HiScores.txt file into the array, separate the data into fields and instantiate the elements of the array. (10)

- 2.3 Code a **toString** method to display all the users each on a separate line. (6)
- 2.4 Code a method to sort the data according to the scores from the highest score to the lowest score. (8)
- 2.5 Code a method to give each user 50 extra points on their score if their time was less than 3 minute (180 seconds). (7)
- 2.6 Some users have their name listed more than once. Code a method to find a user's highest score and then remove all other occurrences of that user from the array. (13)

Application Class

- 3.1 Create an application to do the following. Each time the array is displayed a suitable heading must be given (3)
 - Display the array once it first created (1)
 - Display the array once it is sorted (1)
 - Display the array showing the updated scores (1)
 - Display the array showing each user's unique highest score. (1)

If possible provide a printout of the output.

SQL STATEMENTS

40 marks

Migrate the Access database **ClassicModels** in your T: drive to MySQL
Connect Netbeans to the ClassiModels MySQL database.

A company called Classic Models has stored information about the products it sells, the employees, customers, orders and other information. The create table statements are shown below:

```
CREATE TABLE Customers (
    customerNumber INTEGER NOT NULL,
    customerName VARCHAR(50) NOT NULL,
    contactLastName VARCHAR(50) NOT NULL,
    contactFirstName VARCHAR(50) NOT NULL,
    phone VARCHAR(50) NOT NULL,
    addressLine1 VARCHAR(50) NOT NULL,
    addressLine2 VARCHAR(50) NULL,
    city VARCHAR(50) NOT NULL,
    state VARCHAR(50) NULL,
    postalCode VARCHAR(15) NULL,
    country VARCHAR(50) NOT NULL,
    salesRepEmployeeNumber INTEGER NULL,
    creditLimit DOUBLE NULL,
    PRIMARY KEY (customerNumber)
);
```

```
CREATE TABLE Employees (
    employeeNumber INTEGER NOT NULL,
    lastName VARCHAR(50) NOT NULL,
```

```

firstName VARCHAR(50) NOT NULL,
extension VARCHAR(10) NOT NULL,
email VARCHAR(100) NOT NULL,
officeCode VARCHAR(10) NOT NULL,
reportsTo INTEGER NULL,
jobTitle VARCHAR(50) NOT NULL,
PRIMARY KEY (employeeNumber)
);

CREATE TABLE Offices (
  officeCode VARCHAR(10) NOT NULL,
  city VARCHAR(50) NOT NULL,
  phone VARCHAR(50) NOT NULL,
  addressLine1 VARCHAR(50) NOT NULL,
  addressLine2 VARCHAR(50) NULL,
  state VARCHAR(50) NULL,
  country VARCHAR(50) NOT NULL,
  postalCode VARCHAR(15) NOT NULL,
  territory VARCHAR(10) NOT NULL,
  PRIMARY KEY (officeCode)
);

CREATE TABLE OrderDetails (
  orderNumber INTEGER NOT NULL,
  productCode VARCHAR(15) NOT NULL,
  quantityOrdered INTEGER NOT NULL,
  priceEach DOUBLE NOT NULL,
  orderLineNumber SMALLINT NOT NULL,
  PRIMARY KEY (orderNumber, productCode)
);

CREATE TABLE Orders (
  orderNumber INTEGER NOT NULL,
  orderDate DATETIME NOT NULL,
  requiredDate DATETIME NOT NULL,
  shippedDate DATETIME NULL,
  status VARCHAR(15) NOT NULL,
  comments TEXT NULL,
  customerNumber INTEGER NOT NULL,
  PRIMARY KEY (orderNumber)
);

CREATE TABLE Payments (
  customerNumber INTEGER NOT NULL,
  checkNumber VARCHAR(50) NOT NULL,
  paymentDate DATETIME NOT NULL,
  amount DOUBLE NOT NULL,
  PRIMARY KEY (customerNumber, checkNumber)
);

CREATE TABLE Products (
  productCode VARCHAR(15) NOT NULL,
  productName VARCHAR(70) NOT NULL,
  productLine VARCHAR(50) NOT NULL,
  productScale VARCHAR(10) NOT NULL,
  productVendor VARCHAR(50) NOT NULL,
  productDescription TEXT NOT NULL,
  quantityInStock SMALLINT NOT NULL,
  buyPrice DOUBLE NOT NULL,
  MSRP DOUBLE NOT NULL,

```

```
PRIMARY KEY (productCode)
);

CREATE TABLE ProductLines(
    productLine VARCHAR(50) NOT NULL,
    textDescription VARCHAR(4000) NULL,
    htmlDescription MEDIUMTEXT NULL,
    image MEDIUMBLOB NULL,
    PRIMARY KEY (productLine)
);
```

Write SQL statements in Netbeans and copy the results into the Answer Sheet provided for you in your folder.

1. List the customer's full names who are in Auckland. (3)
 2. Count the number of offices in the USA. (4)
 3. Display the customer numbers whose payments are before 2004 and the amount is over R10000. (5)
 4. Calculate the selling price of the products to be 150 more than their buying price. (5)
 5. Decrease the quantity in stock of the products by 100. (4)
 6. Delete the Employee Gerard Martin. (3)
 7. Find the name of the most expensive product. (4)
 8. Count the average buyprice of the products in each product line. (5)
 9. Find the orders that did not ship in the same month as they were required. Do not display any Cancelled/On Hold/In Process orders. (7)
-