OMNIA LABOR VINCIT

MPUMALANGA DEPARTMENT OF EDUCATION

GRADE 12

INFORMATION TECHNOLOGY P1

Trial Examination - September 2009

MARKS: 120 **TIME: 3 hours**

This question paper consists of 6 questions on 10 pages.

INSTRUCTIONS AND INFORMATION

1. Answer ALL questions in this question paper

2. This is a three hour paper. You will not be permitted to leave the examination room before the end of the examination session.

3. Save at regular intervals as a precaution against power failure.

4. Save ALL your solutions in folders with the question numbers.

5. Type in your name as a comment in the first line and the last line of each program.

6. Read ALL the questions. Do not do more than is required by the question.

7. At the end of the examination session, a soft copy and a hard copy must be handed over to the invigilator/teacher for marking.

8. You require the files listed below in order to answer the following questions. The invigilator will tell you where to find them.

   Question1
   Employee.java
   TestEmployee.java
   SchoolDB.mdb
   Employees.txt
   Funds.txt

   Question2
   Fund.java
   TestFund.java
   Funds.txt

   Question3
   Requisition.java
   TestRequisition.java
   Requisitions.txt

9. You may make use of java API files.You may NOT refer to any other resource material.
10. A soft copy of a programming solution must be handed over to the invigilator.The invigilator will tell you how to send the information.
11. ALL printings of programming questions will take place within an hour of the completion of the examination.

SCENARIO:

Auditors from the Department of Education visited your school to monitor how the school manages the funds. Having realized that the School Management Team (SMT) couldn’t produce the necessary records on how funds were utilized, the auditors recommended that the school should have software to keep all the records on how funds are utilized during the course of the year.

The SMT requested you as a programmer to develop simple programs for the school to manage ALL funds (Petty Cash) utilized.

SECTION A

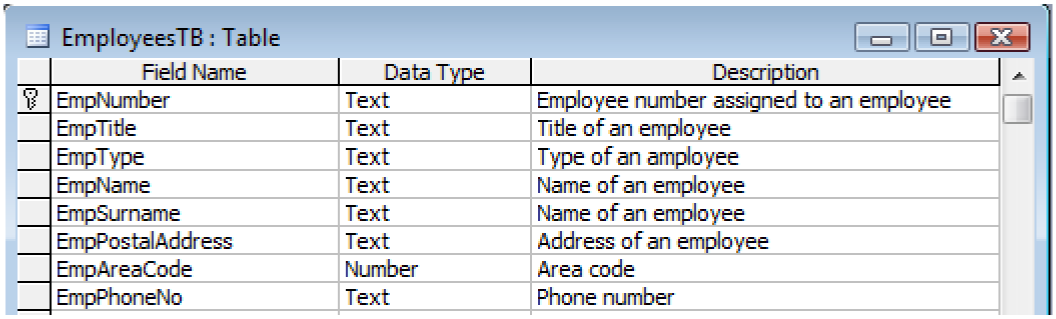
Answer ALL questions in this section

QUESTION 1: JAVA PROGRAMMING AND DATABASES

The records of ALL the staff who received petty cash have been captured in the database named **SchoolDB.mdb**.
The database has been supplied to you in a folder named **Question1.** The database contains two tables named **EmployeesTB** and **FundsTB**.
If you cannot use the database provided, ask the invigilator where to find it or create your own database in the folder named **Question1** by considering the following:

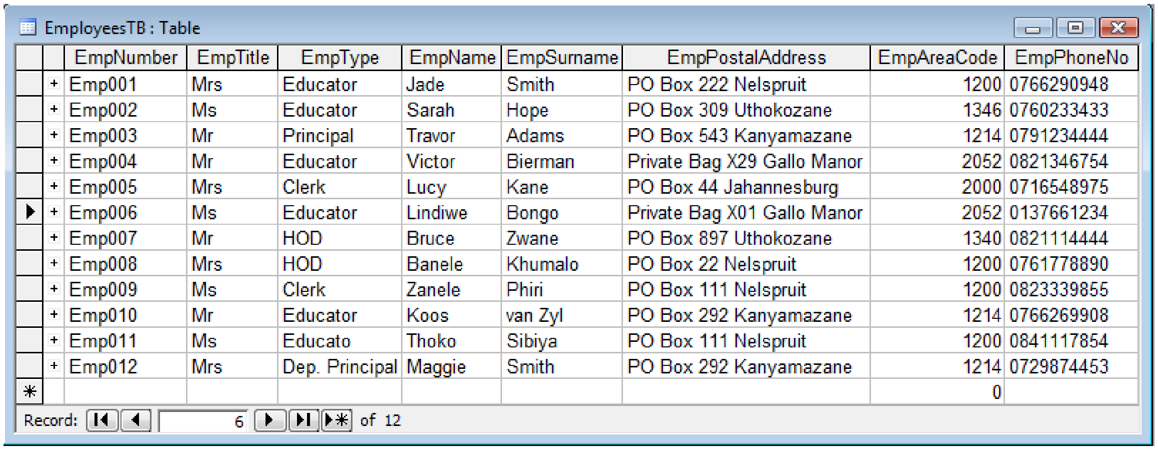
- Use the two text files named **Employees.txt** and **Funds.txt** provided.
- Create tables named **EmployeesTB** and **FundsTB**
- Change the data types and (default sizes) of the fields in the two tables to specification given below. Create a one-to-many relationship (with referential integrity) between the two tables.

EmployeesTB used to store employees record. Fields are defined as follows:



	Field Name	Data Type	Description
🔑	EmpNumber	Text	Employee number assigned to an employee
	EmpTitle	Text	Title of an employee
	EmpType	Text	Type of an amployee
	EmpName	Text	Name of an employee
	EmpSurname	Text	Name of an employee
	EmpPostalAddress	Text	Address of an employee
	EmpAreaCode	Number	Area code
	EmpPhoneNo	Text	Phone number

The following table is an example of the data contained in the **EmployeesTB** table.

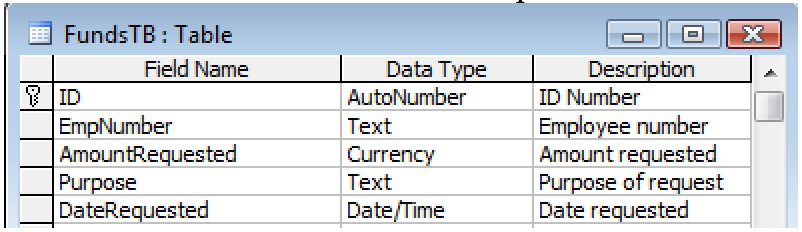


EmployeesTB : Table

EmpNumber	EmpTitle	EmpType	EmpName	EmpSurname	EmpPostalAddress	EmpAreaCode	EmpPhoneNo
Emp001	Mrs	Educator	Jade	Smith	PO Box 222 Nelspruit	1200	0766290948
Emp002	Ms	Educator	Sarah	Hope	PO Box 309 Uthokozane	1346	0760233433
Emp003	Mr	Principal	Travor	Adams	PO Box 543 Kanyamazane	1214	0791234444
Emp004	Mr	Educator	Victor	Bierman	Private Bag X29 Gallo Manor	2052	0821346754
Emp005	Mrs	Clerk	Lucy	Kane	PO Box 44 Jahannesburg	2000	0716548975
Emp006	Ms	Educator	Lindiwe	Bongo	Private Bag X01 Gallo Manor	2052	0137661234
Emp007	Mr	HOD	Bruce	Zwane	PO Box 897 Uthokozane	1340	0821114444
Emp008	Mrs	HOD	Banele	Khumalo	PO Box 22 Nelspruit	1200	0761778890
Emp009	Ms	Clerk	Zanele	Phiri	PO Box 111 Nelspruit	1200	0823339855
Emp010	Mr	Educator	Koos	van Zyl	PO Box 292 Kanyamazane	1214	0766269908
Emp011	Ms	Educato	Thoko	Sibiya	PO Box 111 Nelspruit	1200	0841117854
Emp012	Mrs	Dep. Principal	Maggie	Smith	PO Box 292 Kanyamazane	1214	0729874453
						0	

Record: 6 of 12

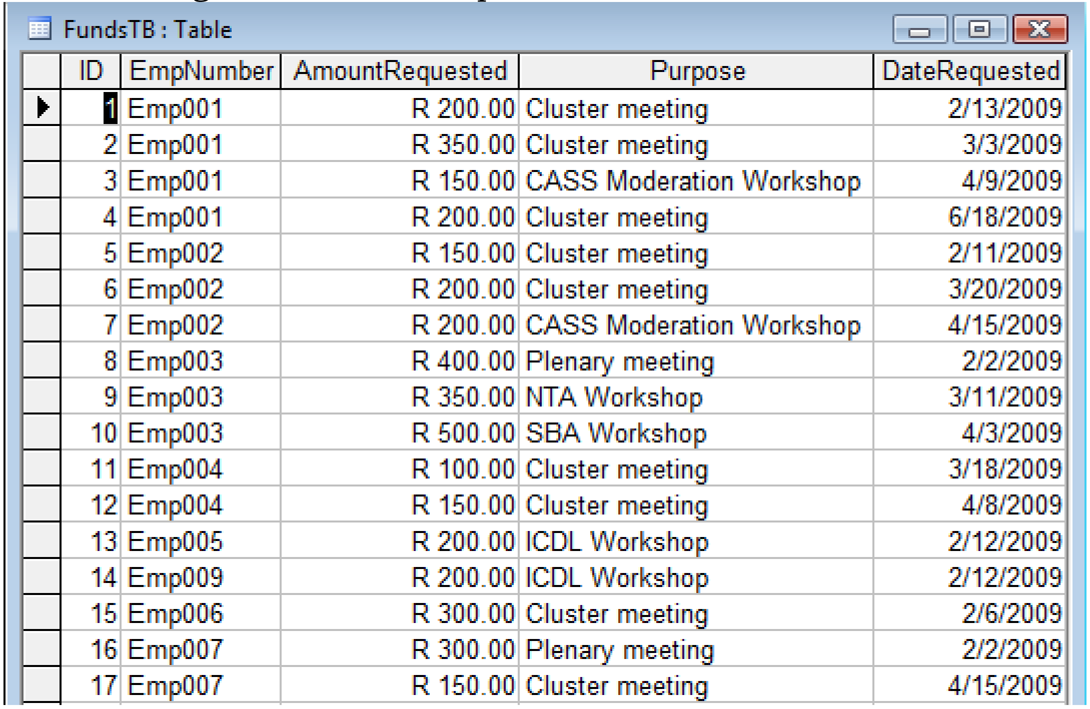
FundsTB used to store funds requested. Fields are defined as follows:



FundsTB : Table

Field Name	Data Type	Description
ID	AutoNumber	ID Number
EmpNumber	Text	Employee number
AmountRequested	Currency	Amount requested
Purpose	Text	Purpose of request
DateRequested	Date/Time	Date requested

The following table is an example of the data contained in the **FundsTB** table.



FundsTB : Table

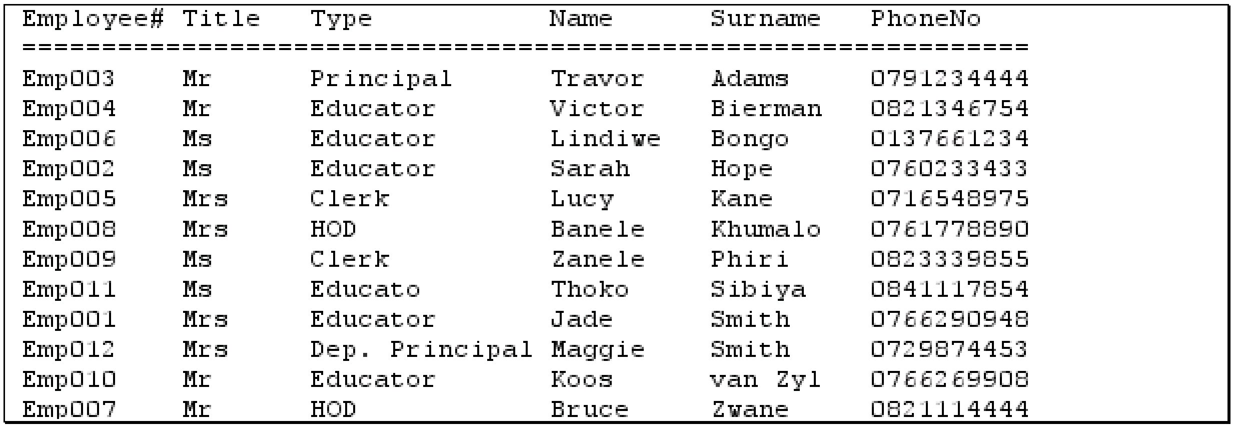
ID	EmpNumber	AmountRequested	Purpose	DateRequested
1	Emp001	R 200.00	Cluster meeting	2/13/2009
2	Emp001	R 350.00	Cluster meeting	3/3/2009
3	Emp001	R 150.00	CASS Moderation Workshop	4/9/2009
4	Emp001	R 200.00	Cluster meeting	6/18/2009
5	Emp002	R 150.00	Cluster meeting	2/11/2009
6	Emp002	R 200.00	Cluster meeting	3/20/2009
7	Emp002	R 200.00	CASS Moderation Workshop	4/15/2009
8	Emp003	R 400.00	Plenary meeting	2/2/2009
9	Emp003	R 350.00	NTA Workshop	3/11/2009
10	Emp003	R 500.00	SBA Workshop	4/3/2009
11	Emp004	R 100.00	Cluster meeting	3/18/2009
12	Emp004	R 150.00	Cluster meeting	4/8/2009
13	Emp005	R 200.00	ICDL Workshop	2/12/2009
14	Emp009	R 200.00	ICDL Workshop	2/12/2009
15	Emp006	R 300.00	Cluster meeting	2/6/2009
16	Emp007	R 300.00	Plenary meeting	2/2/2009
17	Emp007	R 150.00	Cluster meeting	4/15/2009

A folder **Question1** also contains an incomplete java program with two classes named **Employee.java** and **TestEmployee.java**. The program will not run because of the incomplete SQL-statement. Open the incomplete program to make changes. After making changes, the program will display a simple menu with 8 options.

The connectivity code has already been written as part of the given code in the file named **Employee.java**. **COPY** the database from **Question1** folder to **C:\** or specify any location where the database will be located.

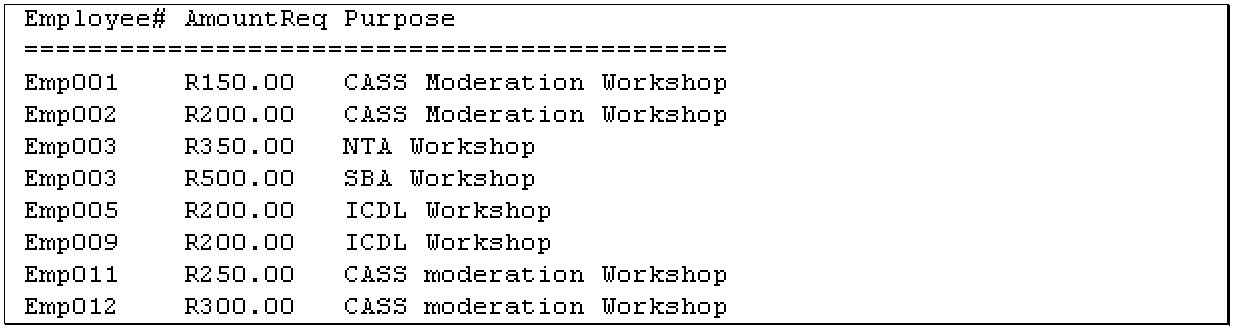
Please **NOTE** that if you can't establish any connection with the database when you execute your program, you must still complete the SQL-statement and submit for marking.

1.1 Create a SQL statement in the **displayAllEmployees()** method to display the **EmpNumber,EmpTitle,EmpType,EmpName,EmpSurname** and **EmpPhoneNo** fields in the **EmployeesTB** table in alphabetical order according to **EmpSurname**. Sample of output. **(4)**



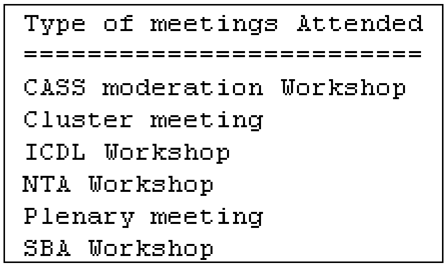
```
Employee# Title   Type           Name      Surname   PhoneNo
===============================================================
Emp003    Mr      Principal      Travor    Adams     0791234444
Emp004    Mr      Educator       Victor    Bierman   0821346754
Emp006    Ms      Educator       Lindiwe   Bongo     0137661234
Emp002    Ms      Educator       Sarah     Hope      0760233433
Emp005    Mrs     Clerk          Lucy      Kane      0716548975
Emp008    Mrs     HOD            Banele    Khumalo   0761778890
Emp009    Ms      Clerk          Zanele    Phiri     0823339855
Emp011    Ms      Educato        Thoko     Sibiya    0841117854
Emp001    Mrs     Educator       Jade      Smith     0766290948
Emp012    Mrs     Dep. Principal Maggie    Smith     0729874453
Emp010    Mr      Educator       Koos      van Zyl   0766269908
Emp007    Mr      HOD            Bruce     Zwane     0821114444
```

1.2 Create a SQL statement in the **displayWorkshops()** method to display all the fields in the **FundsTB** table of all the employees who attended the workshops. Format fields where possible. Sample of output. **(7)**



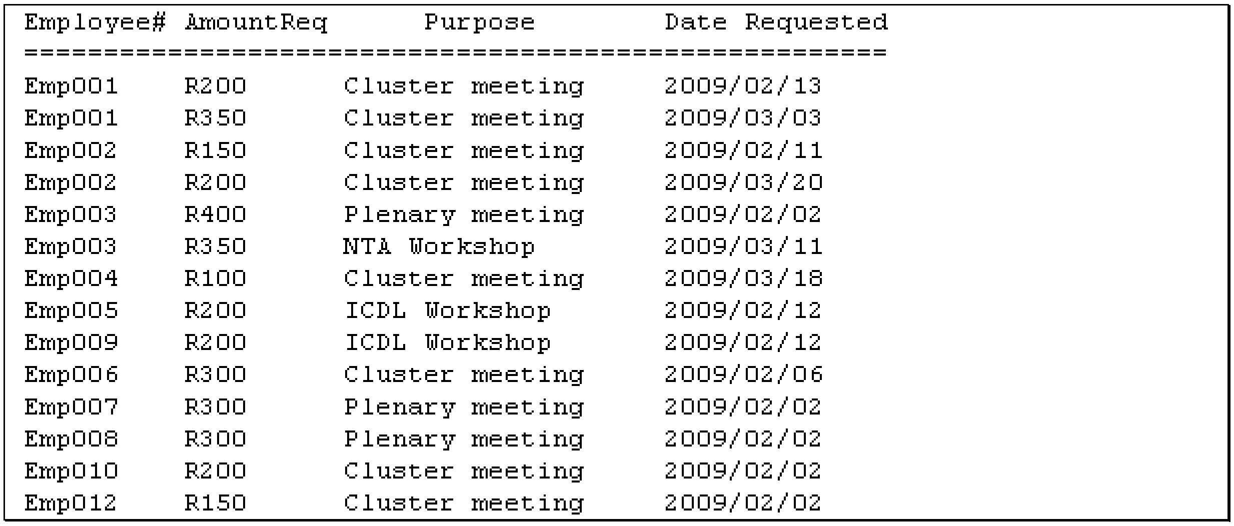
```
Employee# AmountReq Purpose
============================================
Emp001    R150.00   CASS Moderation Workshop
Emp002    R200.00   CASS Moderation Workshop
Emp003    R350.00   NTA Workshop
Emp003    R500.00   SBA Workshop
Emp005    R200.00   ICDL Workshop
Emp009    R200.00   ICDL Workshop
Emp011    R250.00   CASS moderation Workshop
Emp012    R300.00   CASS moderation Workshop
```

1.3 Create a SQL statement in the **displayAllRequests()** method to display a **Purpose** field in the **FundsTB** table of all the employees who requested the petty cash. Only ONE unique purpose should be displayed in the output. Sample of output. **(5)**



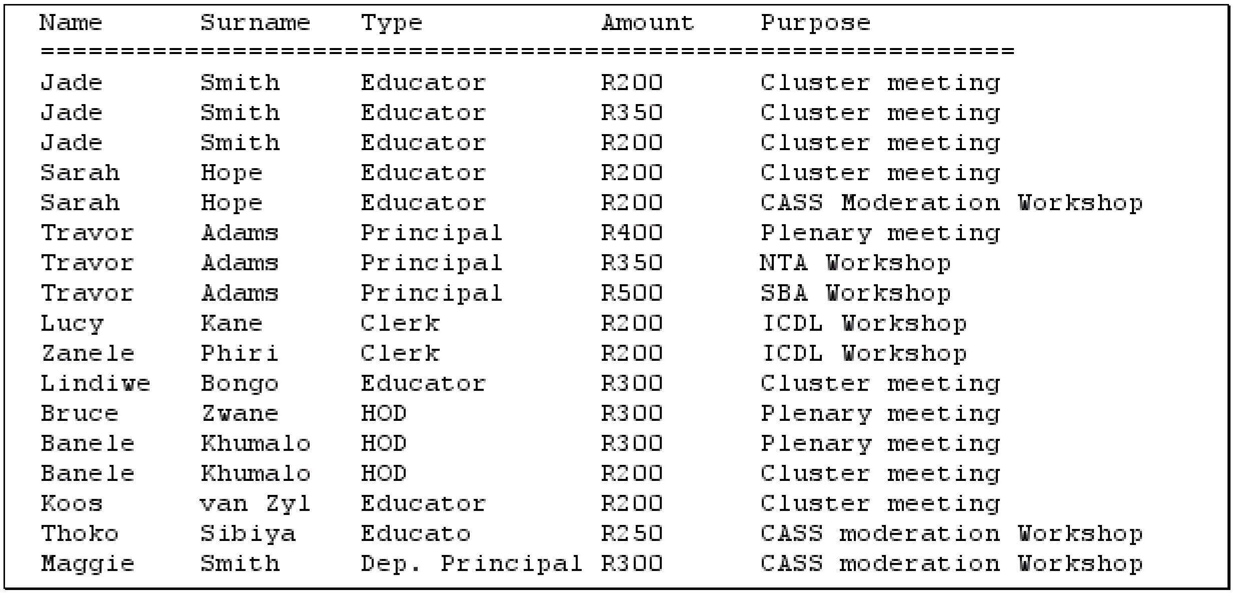
```
Type of meetings Attended
=========================
CASS moderation Workshop
Cluster meeting
ICDL Workshop
NTA Workshop
Plenary meeting
SBA Workshop
```

1.4 Create a SQL statement in the **requestBeforeApril()** method to display the fields in the **FundsTB** table of all the employees who requested the petty cash before april. Sample of output. **(6)**



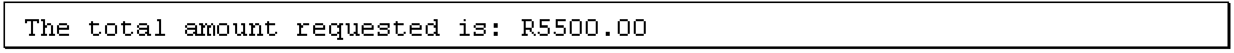
```
Employee# AmountReq      Purpose        Date Requested
======================================================
Emp001    R200      Cluster meeting     2009/02/13
Emp001    R350      Cluster meeting     2009/03/03
Emp002    R150      Cluster meeting     2009/02/11
Emp002    R200      Cluster meeting     2009/03/20
Emp003    R400      Plenary meeting     2009/02/02
Emp003    R350      NTA Workshop        2009/03/11
Emp004    R100      Cluster meeting     2009/03/18
Emp005    R200      ICDL Workshop       2009/02/12
Emp009    R200      ICDL Workshop       2009/02/12
Emp006    R300      Cluster meeting     2009/02/06
Emp007    R300      Plenary meeting     2009/02/02
Emp008    R300      Plenary meeting     2009/02/02
Emp010    R200      Cluster meeting     2009/02/02
Emp012    R150      Cluster meeting     2009/02/02
```

1.5 Create a SQL statement in the **amountRequested()** method to display the **EmpName,EmpSurname,EmpType,AmountRequested** and **Purpose** in both **EmployeesTB** and **FundsTB** tables of all the employees where the amount requested is greater or equals to R200.00. **(7)**



```
Name      Surname   Type           Amount    Purpose
==============================================================
Jade      Smith     Educator       R200      Cluster meeting
Jade      Smith     Educator       R350      Cluster meeting
Jade      Smith     Educator       R200      Cluster meeting
Sarah     Hope      Educator       R200      Cluster meeting
Sarah     Hope      Educator       R200      CASS Moderation Workshop
Travor    Adams     Principal      R400      Plenary meeting
Travor    Adams     Principal      R350      NTA Workshop
Travor    Adams     Principal      R500      SBA Workshop
Lucy      Kane      Clerk          R200      ICDL Workshop
Zanele    Phiri     Clerk          R200      ICDL Workshop
Lindiwe   Bongo     Educator       R300      Cluster meeting
Bruce     Zwane     HOD            R300      Plenary meeting
Banele    Khumalo   HOD            R300      Plenary meeting
Banele    Khumalo   HOD            R200      Cluster meeting
Koos      van Zyl   Educator       R200      Cluster meeting
Thoko     Sibiya    Educato        R250      CASS moderation Workshop
Maggie    Smith     Dep. Principal R300      CASS moderation Workshop
```

1.6 Create a SQL statement in the **totalAmountRequested()** method to display the total amount requested by employees in the school. The results should be part of a message. Sample of output. **(6)**



```
The total amount requested is: R5500.00
```

Hint: Use **Amountrequested** field in the **FundsTB** table to calculate the petty cash.

1.7 Complete the code in the **deleteEmployeeRecord()** by asking the user to enter the employee number. Create a SQL statement to delete a record in the **EmployeeTB** table given an employee number. **(5)**

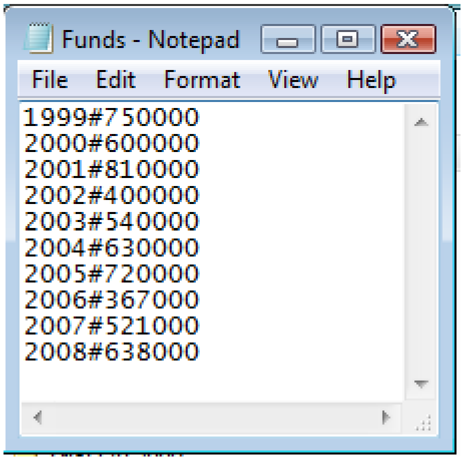
In **TestEmployee.java** class, invoke (call statement) the relevant methods to run your program according to the given options.

TOTAL MARK: 40

QUESTION 2: OBJECT – ORIENTED PROGRAMMING

The following questions focus on Object – Oriented Programming. Include ALL the classes given in the instructions.

The funds the school has raised from 1999 to 2008 are stored in the text file named **Funds.txt** in the following format.



```
1999#750000
2000#600000
2001#810000
2002#400000
2003#540000
2004#630000
2005#720000
2006#367000
2007#521000
2008#638000
```

Do the following:

- Create two classes called **Fund.java** and **TestFund.java**
- Open **Question2** folder
- Open **Fund.java** and **TestFund.java** classes.
- Copy and paste the data from the given incomplete program to the classes created above.

2.1 Write the following code in the **Fund.java** class.

2.1.1 This class must contain the following fields.

funds
year

Ensure that you choose appropriate data types for these fields. (2)

2.1.2 Write a parameterized constructor that will pass values for ALL the fields in the class. These parameters should be used to initialize a new object. (2)

2.1.3 Write mutator (set) and accessor (get) methods for ALL the values in this class. (4)

2.1.4 Write a method of type string named **toString** that returns information on school funds formatted as follows. Format the funds field to two decimal places after a comma.

(4)

Funds, year
Example of return string for the school funds
R750000.00 1999

2.2 Do the following in the **TestFund.java** file in the given program.

2.2.1 Create an array of object called **funds** that keeps school funds records. Set the size of the array to 20.

(2)

2.2.2 Write code to read information from the **funds.txt** text file according to the following steps.

(a) Test if the file exists. If it exists, initialize a loop to read data. Display a suitable message if the file does not exist and terminate the program.
(b) Read a line of text from a text file.
(c) Check if the line from the text file is not an empty string
(d) If the line is not empty, separate the text into the **funds** and the **year** in which the funds were received.
(e) Use the information to create a new **funds** object and place the object into the array.
(f) Close file.

(12/2 = 6)

2.2.3 Write a code to complete the options on the menu provided in the program. The methods in the **Fund.java** class should be used where applicable. Call the relevant methods from the class.

Print ALL funds: Display ALL fields from the array. Call the **toString()** method to display the information. Display suitable heading.

(4)

Print ALL funds greater than R600000.00: Display ALL the fields if a fund is greater than R600000.00. Call the **toString()** method to display the information. Display suitable heading.

(2)

Print Total funds raised between the year 2000 and 2005: Display ALL the fields if the year is between 2000 and 2005 (2000 and 2005 included). (6)

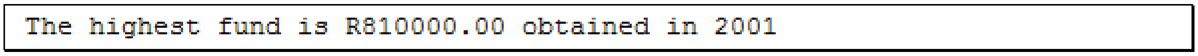
Add new funds: Ask a user to input **funds** and the **year**. Use the information to create a new object. The new object must be added into an array and the information must also be added to the **Funds.txt** text file.

(8/2 = 4)

Sort According to Funds in descending order: Display a sorted array according to funds in descending order. Call a **toString()** method to display the information. Use the suitable methods from the **Fund.java** class to get the results. Display a suitable heading.

(8)

Print the Highest Fund: Display the fields of the object with the highest fund. Display the results in a message. Sample of output.



```
The highest fund is R810000.00 obtained in 2001
```

(6)

TOTAL MARK: 50

QUESTION 3: PROGRAMMING

This question consists of three methods. Use **Question3** folder to answer ALL the questions.

3.1. At the beginning of each year the head master requests all the head of departments (HOD) to submit their requisitions. The list of all the HOD's, departments and their requisitions is declared in an array called **request** in the class called **Requisition.java**.

Do the following :

- Create two classes called **Requisition.java** and **TestRequisition.java**
- Copy and paste data from the incomplete program to your classes.
- Complete the code by answering the following questions.

Do the following in the **Requisition.java** class:

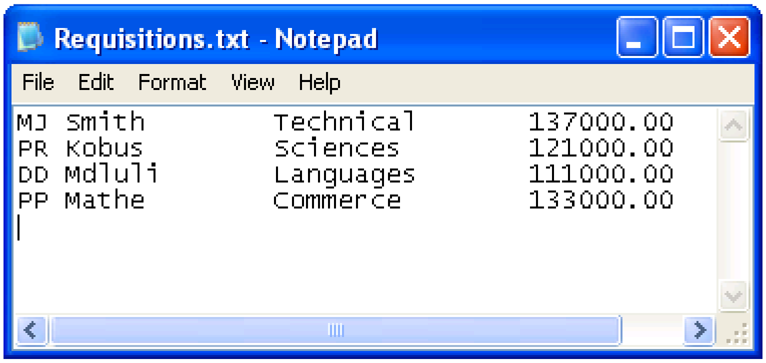
3.1.1 Create a method named **displayDepartments()** to print ALL the school departments and their HOD's. Display suitable headings.

(5)

3.1.2 Create a method named **totalRequisitions()** to calculate the total requisitions made by all the departmental heads. Display the total requisitions as part of a message.

(7)

3.1.3 Create a method named **saveRecords()** to store all the records from the array to the text file called **Requisitions.txt** in the **Question3** folder. Example of the text file:



```
MJ Smith          Technical        137000.00
PR Kobus          Sciences         121000.00
DD Mdluli         Languages        111000.00
PP Mathe          Commerce         133000.00
```

(10)

3.2 In **TestRequisition.java** class, invoke (call statement) the relevant methods to run your program according to the given options.

(8)

TOTAL MARK: 30
GRAND TOTAL: 120