higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

NATIONAL CERTIFICATE (VOCATIONAL)

PRINCIPLES OF COMPUTER PROGRAMMING

(Second paper**)**

NQF LEVEL 3

(10041023)

12 March 2020 (X-paper)
09:00–13:00

CANDIDATES ARE ALLOWED 15 MINUTES TO READ THE INSTRUCTIONS BEFORE THE START OF THE EXAMINATION.

This question paper consists of 17 pages.

551Q2S2012

Copyright reserved

TIME: 4 HOURS
MARKS: 100

INSTRUCTIONS AND INFORMATION

1. Candidates must be seated 30 minutes before the start of the examination to ensure that the computers have the appropriate software and can make PRINTOUTS.

2. Answer all the questions.

3. Read all the questions carefully.

4. Number the answers according to the numbering system used in this question paper.

5. This question paper is set with programming terms that are specific to the BASIC programming language and must be completed using Visual Studio.NET or VB.NET Express as well as MS Office or any other applicable word processing software.

6. Routines such as *search*, *sort* and *selection* must be hard-coded by the candidate. The built-in features of the programming language must not be used.

7. All data structures must be defined unless the question has a partially completed source code.

8. The partially completed source code must not be deleted.

9. The source code must be printed directly from the integrated development environment (IDE). Candidates are not allowed to copy the source code from the IDE and paste it to MS Word for printing purposes.

10. All PRINTOUTS of the source code and screen prints must be handed to the invigilator.

11. Candidates may use the help function of the integrated development environment (IDE).

12. Save the work regularly on the hard disk/flash, disk/network drive in use to prevent loss of work. No additional time will be allowed for work lost due to lack of saving.

Copyright reserved

13. In the event of problems such as a power failure or a computer or printing device breakdown the invigilator will make the necessary arrangements so that the candidate is able to continue with the examination session. Candidates will not be penalised with time lost under such circumstances.

14. No candidate may print his or her work or make a flash drive available for another candidate or access another candidate's work via the network. Any attempt to access any information from or transfer information to another candidate in whatever form is a contravention of the examination rules and regulations and will be viewed in a serious manner.

15. Print the answer after a question has been completed. The candidate's EXAMINATION NUMBER must appear on all PRINTOUTS. No PRINTOUTS without a printed EXAMINATION NUMBER will be marked.

16. Steps to print a user interface or any other screen print:

    16.1 Run/Debug the solution (for Visual Studio.NET only).

    16.2 Hold down the ALT key and press the Prnt Scrn key.

    16.3 Open a blank MS Word document and paste the screen print.

    16.4 Insert a blank THREE columns header.

        (a) Add your EXAMINATION NUMBER against the left margin.

        (b) Add the QUESTION NUMBER against the right margin.

    16.5 Make a PRINTOUT of the page.

17. The following VB.NET solution files have been saved on the computer in the PCP L3 P2 Examination Year folder:

    17.1 QUESTION 1

File name	Folder name
VehicleRental.sln	VehicleRental

    17.2 QUESTION 2

File name	Folder name
CarCare.sln	CarCare

Copyright reserved

QUESTION 1: FUNCTION AND SUBPROCEDURES

The following questions refer to the VB.NET solution *VehicleRental* that has been saved on the computer.

A vehicle rental company rents motor vehicles to clients. The rental manager has the responsibility to check if a vehicle is available or not when a client requests to hire it. If the vehicle is available, a rental form is completed and the rental details are stored in a rental file. The company's vehicle fleet is divided into categories as indicated in TABLE 1 below.

VEHICLE CATEGORY	VEHICLE MAKE	DESCRIPTION	STATUS/AVAILABILITY	PRICE/DAY
Sedan				
	Volvo S60	Premium auto sedan	Sold out = 2	R948
	Mercedes C-Class	Luxury auto sedan	Available = 1	R950
Hatch				
	Ford Fiesta	Manual hatch	Not available = 0	R418
	Audi A3	Automatic hatch	Available = 1	R671
Compact hatch				
	Hyundai i10	Standard compact	Available = 1	R293
	Ford Figo	Large compact	Available = 1	R329

TABLE 1: VEHICLES FOR HIRE

You have to design an application that allows customers to do the following:

- Choose a vehicle
- Complete the rental form
- Create a booking
- Display the rental confirmation

NOTE: In order to secure the booking the company requires a credit card balance that is double the amount of the rental price.

Copyright reserved

Below is the IPO diagram for the problem.

INPUT	PROCESSING	OUTPUT
Vehicle type Vehicle make Client name Client ID number Credit card balance Rental period (in days)	Amount due = Rental fee * Number of days	Vehicle name Rental period (days) Total rental cost

FIGURE 1 below shows a screen print (screenshot) of the graphical user interface:

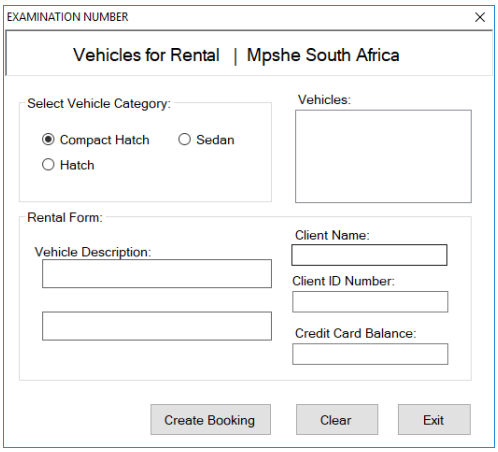
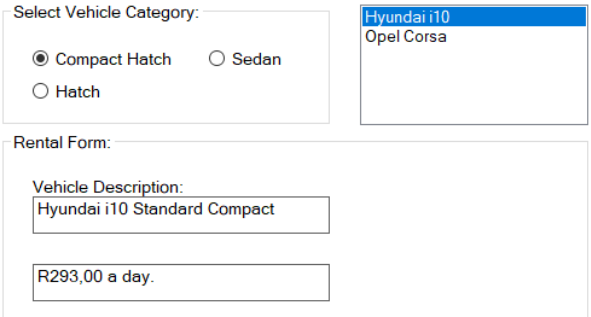


FIGURE 1: VEHICLE RENTAL USER INTERFACE

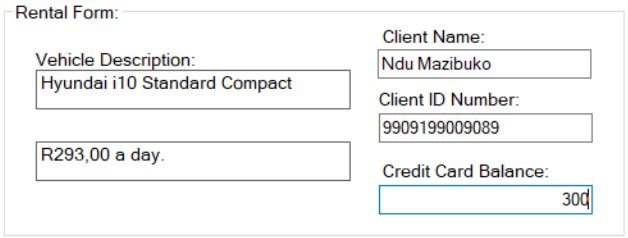
The steps below explain how the user selects a vehicle, completes the rental form and produces the rental confirmation:

STEP 1 Run the application. Select the *Compact Hatch* and *Hyundai i10* as shown below. The application will display the relevant information.

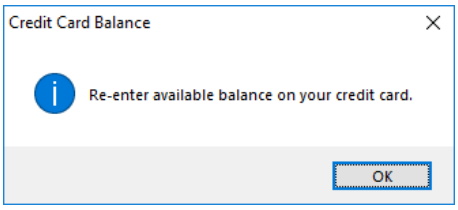


Copyright reserved

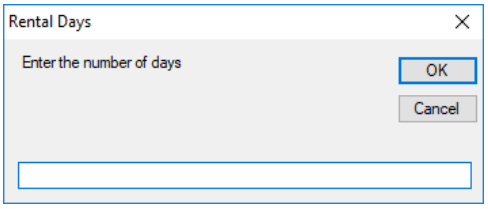
STEP 2 Enter the client name, ID number, current available credit card balance and click the *Create Booking* button.



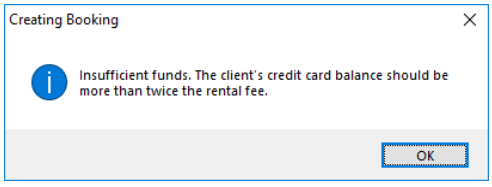
STEP 3 If the credit card balance is not entered, the application will alert the user as shown below.



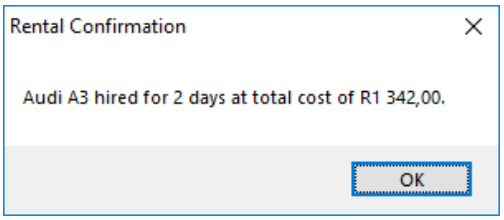
STEP 4 When all relevant data has been entered and the user clicks the *Create Booking* button, the application will prompt him to enter the rental period (in days).



STEP 5 If the client's current credit card balance is less than double the rental cost, the rental request will be declined and the user will be alerted with an appropriate message.



STEP 6 If the credit card balance is enough, the rental confirmation will be issued as shown below.



Copyright reserved

The completed user interface with partially completed source code has been saved on the computer. Open the *VehicleRental* solution and follow the instructions to complete the application.

1.1 Add the following IDE settings and form class declarations:

1.1.1 Ensure that option strict is enforced. (1)

1.1.2 Declare and appropriately initialise a variable named *intClientCount* to count the number of clients. (1)

1.1.3 Declare and appropriately initialise a variable named *intStatus* to indicate the status (availability) i.e. 0 or 1 or 2 of the vehicle. (1)

1.1.4 Declare and appropriately initialise a variable named *decRentalfee* to store the applicable rental cost. (1)

1.2 Using a *restricted* access modifier, create a user defined subprocedure called *PopVehicleNames* without parameters which will perform the following: (2)

1.2.1 Remove all items from the *Vehicle* list box. (1)

1.2.2 (a) Use a nested *If ElseIf* statement to determine which *category* of *vehicle* has been selected.

(b) And then populate the *Vehicle* list box with the appropriate names of the vehicles based on that *category*.

HINT: Use the global *String* variables that have already been declared and assigned values. (5)

1.2.3 Select the first item in the *Vehicle* list box. (1)

1.3 Using a *restricted* access modifier create a user-defined subprocedure called *DisplayVehicleInfo* which will display vehicle information using the following instructions:

1.3.1 The subprocedure must accept as a copy the following THREE parameters as a copy.

- Description
- Rental fee
- Availability (status) (2)

1.3.2 Concatenate the current vehicle name from the list box together with the value of the *Description* parameter and assign the value to the *Vehicle Description* label control. (2)

1.3.3 Assign the value from the *Availability* parameter into the flag variable named *intStatus*. (1)

1.3.4 Disable the *Create Booking* button. (1)

Copyright reserved

1.3.5 (a) Use an *If ElseIf* statement to determine if the vehicle is *not available* or if it is *sold out*, whilst assigning an appropriate message to the *Rental Fee* label control;

(b) Otherwise
- enable the *Create Booking* button
- assign the value from the *Rental fee* parameter into the variable named *decDailyRentalFee*
- display the rental fee formatted as currency with TWO decimals

HINT: See STEP 1

NOTE: QUESTION 1.3.5 (a) and QUESTION 1.3.5 (b) use a single *conditional* statement. (4)

1.4 Add program code to the *Checked changed* event of every radio button. Determine if the radio button is checked, and then invoke the subprocedure created in QUESTION 1.2 above. (2)

1.5 Using a *restricted* access modifier, create a user-defined function procedure called *IsBalanceOk* and add code to perform the following:

1.5.1 The function procedure must accept as a copy ONE parameter named *Amount* and must return a flag value. (3)

1.5.2 Determine if the entered *credit card balance* is more than twice the value of the parameter named *Amount* or not and return an appropriate flag value in each case. (3)

1.6 A user-defined subprocedure named *SaveRentalData* with TWO value parameters (amount and days) has been partially completed. Complete the rest of the program code to perform the following:

1.6.1 Change the access modifier of the subprocedure to restrict access. (1)

1.6.2 Increment the *client* counter variable by 1. (1)

1.6.3 Create a pop-up dialog box to display the rental confirmation note as shown in STEP 6 above. (2)

1.7 Add the following program code to the Click Event of the Create Booking button:

1.7.1 Create an input dialog box to request the user to enter number of days and assign the value into a local variable (numberOfdays). (2)

1.7.2 Calculate the amount due. (1)

1.7.3 Determine whether the client's *credit card balance* is sufficient enough or not by adding the program code to perform the following:

Copyright reserved

(a) Invoke the function procedure named IsBalanceOK and assign its returned value into an appropriate local *Boolean* variable.

HINT: Use an appropriate local variable as an argument. (1)

(b) If the balance is insufficient, alert the user, focus the *Credit Card Balance* textbox control and abort the procedure.

HINT: See STEP 5 above. (3)

(c) However, if the balance is sufficient, then invoke the following subprocedures:

- SaveRentalData
- ChangeVehicleStatus
- ClearRentalForm

NOTE: Include and use the arguments/parameters where applicable. (2)

1.8 Insert your EXAMINATION NUMBER in the source code and print.

1.9 Change the text property of the form to your EXAMINATION NUMBER.

Run the application and enter the following test data:

```
Select vehicle type :  Hatch
Vehicle make        :  Audi A3
Client name         :  Lentsoane Sekola
Client ID number    :  7902294908088
Credit card balance :  R5500.00
Number of days      :  2
```

Click the Create Booking button and make a screen print of the output and print. (1)

[45]

QUESTION 2: ARRAYS

The questions refer to the VB.NET solution named *CarCare* that has been saved on the computer.

A group of students acquired car wash equipment from an NGO and started small operations in a newly established shopping mall.

They offer the following major services to clients:

- Car wash
- Auto detailing (valet services)
- Engine and/or chassis steam cleaning

You have to create a computer program that will assist them to produce a tax invoice for their clients. The program should allow the user to enter the customer name, contact number, car's registration number, make and model, colour and the category of the vehicle.

Prices are as follows:

Car wash options (including VAT)	Sedans to 4x2s	4x4s and minibus
Wash and go	R60	R70
Wash with tyre shine	R70	R80
Engine and chassis	R300	R325
Auto detailing mini	R500	R550

TABLE 2: CAR WASH OPTION PRICES

Extra options (including VAT)	Sedans to 4x2s	4x4s and minibus
Drying	R 10	R 15
Interior cleaning and vacuum	R 15	R 20

TABLE 3: EXTRAS PRICES

FIGURE 2 below is a screen print (screenshot) of the Car wash user interface.

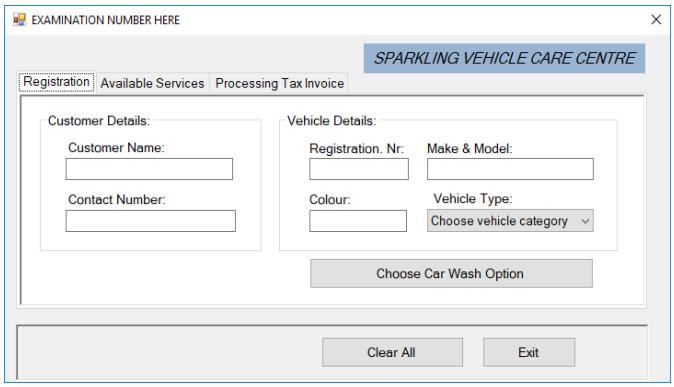
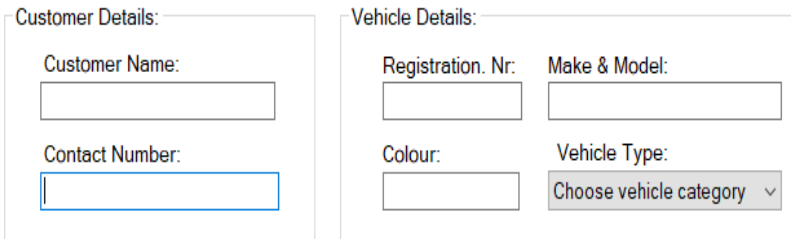


FIGURE 2: CAR WASH USER INTERFACE

Copyright reserved

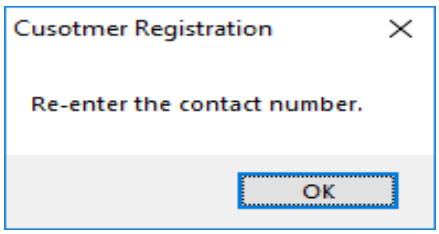
The steps below explain how to use the car wash application to display the client's tax invoice.

STEP 1 Run the application and enter the customer name, contact number, registration number, make and model, colour and the vehicle type.

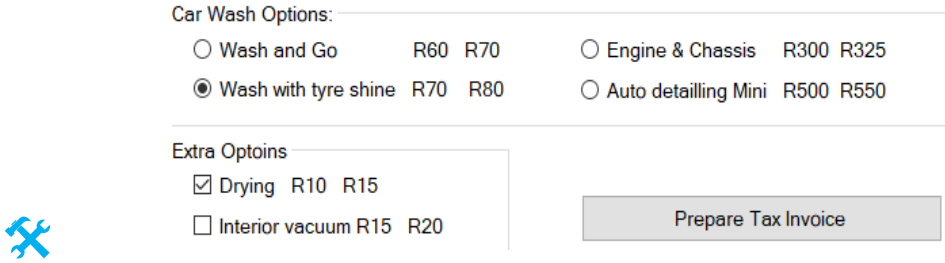


STEP 2 Click on the Choose Car Wash Option button. If the mandatory field is not completed, the system alerts the user with an appropriate message.



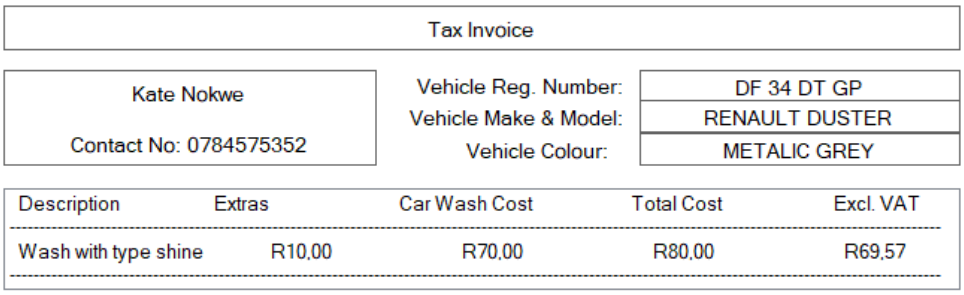


STEP 3 Click on the Wash with tyre shine radio button and tick in the Drying checkbox button as shown below.



Copyright reserved

STEP 4 Click on the Prepare tax invoice button and the system will produce the tax invoice as shown below.



Tax Invoice

Kate Nokwe Contact No: 0784575352		Vehicle Reg. Number: Vehicle Make & Model: Vehicle Colour:	DF 34 DT GP RENAULT DUSTER METALIC GREY

Description	Extras	Car Wash Cost	Total Cost	Excl. VAT
Wash with type shine	R10,00	R70,00	R80,00	R69,57

The completed user interface with partially completed source code has been saved on the computer.

Open the CarCare solution and follow the instructions to complete the application.

2.1 Declare the following at form level:

2.1.1 A 2-dimensional array called *decArrCarWashPrices*. The first, second and third rows contain zeroes, sedan prices and 4x4 prices respectively. Refer to TABLE **2** above for information. (2)

2.1.2 A 2-dimensional array called *decArrExtrasPrices*. The first, second and third rows contain zeroes, sedan prices and 4x4 prices respectively. Refer to TABLE **3** above for information. (1)

2.1.3 THREE variables to store the *car wash cost*, *extras cost* and *total cost* respectively. (1)

2.2 Using an open-access modifier create a subprocedure called *DisplayCustomerInfo* without parameters. The procedure must perform the following: (2)

2.2.1 Concatenate the customer name and contact number and assign it to the *lblCustomerDetails* display label. (2)

2.2.2 Assign the appropriate values to the *lblRegNumber*, *lblMakeModel*, and *lblColour* display labels. (1)

2.3 Add program code to the click event of the *Choose Car Wash Option* button on the registration tab to display the customer information.

2.3.1 Determine if the entered contact number is valid or not by invoking the function procedure named *ValContactNumber*. If not, then alert the user with an appropriate message, set the focus to the contact number textbox and abort the procedure. (4)

Copyright reserved

2.3.2 Determine if the entered registration number is valid or not by invoking the function procedure named *ValTextField* to If not, alert the user with an appropriate message, set the focus to registration number textbox and abort the procedure. (1)

HINT: For QUESTION 2.3.1 and QUESTION 2.3.2, use a conditional statement with the procedure.

2.3.3 Check if a vehicle type had been selected. If not, then alert the user with an appropriate message set the focus to the vehicle category's combo box and abort the procedure. Otherwise invoke the subprocedure created in QUESTION 2.2 and set the tab of the tab control to the *Available services* tab page. (4)

2.4 Add the following program code to the user-defined subprocedure named *CalcTotalCost:*

2.4.1 Assign the car wash price from the two-dimensional array created in QUESTION 2.1.1 to an appropriate variable.

HINT: Use the *SelectedIndex* of the Vehicle type combo box for the row and the variable *colIndex* for the column of the array element. (3)

2.4.2 If the drying checkbox is ticked, then assign the *extra cost* price from the two-dimensional array created in QUESTION 2.1.2 to the appropriate variable.

HINT: Use the *SelectedIndex* of the Vehicle type combo box for the row and the first column of the array element. (3)

2.4.3 If the vacuum checkbox is ticked, add the extra cost price calculated in QUESTION 2.4.2 to the extra cost price from the two-dimensional array created in QUESTION 2.1.2 and assign it to an appropriate variable.

HINT: Use the *SelectedIndex* of the Vehicle type combo box for the row and the second column of the array element. (2)

2.4.4 Calculate the total cost by adding the car wash price and extra cost price. (1)

2.5 A user-defined function procedure named *CalcCostExcludingVAT* has been partially completed.

NOTE: This function procedure receives the total amount including VAT and will return the total amount excluding VAT.

Complete the rest of the program code using the following instructions:

2.5.1 Change the function procedure's definition to accept, as a copy, a decimal-type parameter named *amount*.

Copyright reserved

2.5.2 Change the return type of the function to *decimal*.

2.5.3 Calculate and return the total excluding the VAT. The formula is the amount divided by 1.15.

(3 × 1) (3)

2.6 Test and debug the source code.

2.7 Insert your EXAMINATON NUMBER into the source code and make a PRINTOUT.

[30]

QUESTION 3: DATABASE

You have to design a database application for a new recruitment agency so that they can keep track of client details and available jobs.

The database design selected consists of two tables and a one-to-many relationship between them.

Below is the data specification of the two tables.

FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
ClientID	Text	4	4 characters unique code Primary Key(PK)
ClientName	Text	20	Client's name
ClientContactNo	Text	12	Client's contact number

TABLE 4: CLIENT

FIELD NAME	DATA TYPE	FIELD SIZE	LOOKUP/VALIDATION	DESCRIPTION
JobID	Text	6	n/a	6 characters unique code (PK)
JobTitle	Text	40	n/a	Description of the job
Area	Text	20	n/a	The place where duties will be performed.
AnnualSalary	Number	Decimal	>=150 000	Job offer
ClientID	Text	4	ClientID	Foreign key

TABLE 5: JOB

The application needs the following data relation:

A relationship between two tables using the client ID as the key (1: M).

Copyright reserved

3.1 Use the information above to set up the database in MS Access. Open a new blank database and save it as *Jobs* with your EXAMINATION NUMBER: *Jobs – Examination number*.

3.1.1 Create a Clients table (refer to TABLE 4) and save it as CLIENT. (3)

3.1.2 Create a Jobs table (refer to TABLE **5**) and save it as JOB. (3)

3.1.3 Create a 1:M relationship between CLIENTS and Jobs tables. Use the ClientID field as key. Enforce referential integrity. (2)

3.1.4 Open the Jobs table. Select the Salary field.

Make a screen print and print the table design view. (1)

3.1.5 Use the database documenter to set the print table definition options as shown in FIGURE 3 below.

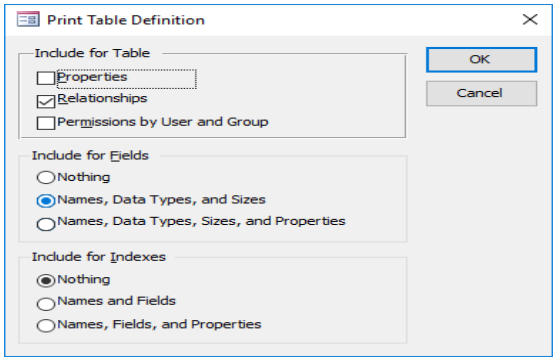


FIGURE 3: TABLE STRUCTURE SETTINGS

Print the database table structures for all tables.

3.2 Create a database form based on Jobs table.

Use the form wizard to create a simple form based on the Jobs table. Format the header as shown in FIGURE 4 below and select all the fields from the Jobs table.

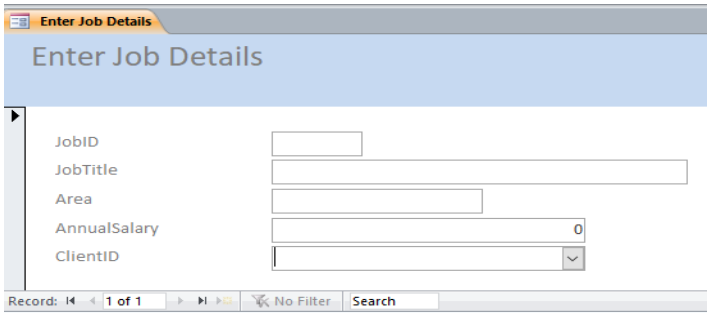


FIGURE 4: JOB DETAILS FORM

Make a screen print of the form and print. (2)

3.3 Follow the instructions below to capture all input data into database tables:

3.3.1 Open the database table called Clients in a datasheet view mode and insert the following records:

CLIENT ID	CLIENT NAME	CONTACT NUMBER
1001	ABB South Africa	011 567 8000
1002	Discover Inkwe Insurance	015 645 6000
1003	TeleClient (PTY) Ltd	013 456 3434

3.3.2 Open the database form that was created in QUESTION 3.2.1 and insert the following records:

JOB ID	JOB TITLE	AREA	ANNUAL SALARY	CLIENT ID
000001	Mechanical Fire Safety Engineer	Centurion	R224 000	1001
000002	Sales admin assistant	Boksburg	R188 000	1002
000003	Project Manager: Legal Services	Johannesburg	R300 000	1001
000004	Customer Care Consultant	Centurion	R156 450	1003
000005	Senior Financial Accountant	Centurion	R402 500	1001
000006	Assistant Control Engineer	Springs	R290 500	1002
000007	Mechanical Draughts Person	Cape Town	R210 450	1003

3.3.3 Re-enter the annual salary for Job ID 000002 using the correct amount R88 000.

Make a screen print of the error dialog box and print. (1)

3.4 Create a database query based on the JOB table using the instructions given below:

3.4.1 The query must retrieve data from the following fields:

JOB table (JobID, JobTitle, and Area) (1)

3.4.2 Add a new field called Monthly Salary and calculate the monthly salary. (3)

3.4.3 Add a condition to retrieve all jobs from the *Centurion* area. (1)

3.4.4 Open the query in design view. Ensure that all columns are clearly legible. Make a screen print of the query design and print it.

Copyright reserved

3.4.5 Run the query, select all records and print the selected records. (1)

3.5 Create a database report based on the query created in QUESTION 3.4 above. Follow all the instructions given below:

3.5.1 The report must display all the fields from the query. (1)

3.5.2 Sort by Job title. (1)

3.5.3 Report layout: Tabular and Portrait.

3.5.4 Report title: Selected Vacancies with your EXAMINATION NUMBER. (1)

3.5.5 Add a formula to determine the lowest monthly salary. (2)

3.5.6 Add Lowest as a descriptive label to the formula created in QUESTION 3.5.5 above. (1)

3.5.7 Add the currency symbol and TWO decimal places to the lowest monthly salary. (1)

3.5.8 Print the report.

[25]

TOTAL: 100

Copyright reserved