

Visual Basic R Graphics Programming Hands On Applications And Advanced Color Development 2nd Edition

This book is designed to introduce programmers to programming and computational thinking through the lens of exploring database. This book offers Python programmers one place to look when they need help guiding to Python as one of the fastest-growing computer languages including Web and Internet applications. This clear and concise introduction to the Python language is aimed at readers who are already familiar with programming in at least one language. This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MariaDB and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MariaDB and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create and configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter nine, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

All the tools you need to create the full range of Visual Basic(r) color graphics applications Expert Rod Stephens provides you with everything you need to add advanced graphics to your applications in this in-depth introduction to graphic programming with Microsoft Visual Basic. From images using as few as 16 colors to "true-color" applications that use more than 16 million, he shows you how to create the full range of color graphics applications. You'll learn how to use Visual Basic controls to create impressive graphic effects without having to buy expensive add-on products. This book/CD-ROM package also explains how to integrate imaging, animation, and two- and three-dimensional graphics into an application. And you'll find the tools to manipulate color images, overlay one image on another, build scrolled windows, and much more. The Second Edition covers: * New API functions * Bitmap image morphing * New algorithms for hidden surface removal * Print preview with multiple pages and scales * Image processing, including high color and true color * Examples of controlling animation using simulation * New examples that demonstrate shape-distorting transformations * New examples of fractals and tilings * Gouraud shading, Phong shading, and texturing * Ray tracing speed improvements * Ray tracing for new kinds of objects The CD-ROM includes: * More than 400 complete, ready-to-run example programs * Pictures to use with the example programs * Images generated by the programs * Color images of many of the figures from the book * Source code for all example programs from the First Edition

Visual Basic(R) Programming "Addressing various business applications, the authors discuss the programming topics needed to implement those applications and provide simple, short, complete examples. Chapter 2 emphasizes how quickly readers can design Windows-based user interfaces." -- Narges Kasiri, Oklahoma State University "A well written, authoritative textbook for all those who want to learn Visual Basic programming from the ground up." -- Dr. Hamid R. Nemat, The University of North Carolina at Greensboro "Contains the information a non-programmer needs to become a master Visual Basic programmer. Excellent code examples." -- Jeffrey P. Scott, Blackhawk Technical College "This book teaches you everything you need to know to build great applications the right way." -- Joe Stagner, Microsoft More than a million students and

professionals have learned programming and software development with Deitel(R) "How to Program" series books. "Visual Basic 2010 How to Program" provides a clear, simple, concise and engaging late objects introduction to Visual Basic programming. "Features: " Completely rewritten to focus on introductory Visual Basic courses. Rich coverage of fundamentals, including two chapters on control statements. Focus on real-world business examples. New Making a Difference exercises set. Integrated Windows Forms GUI, debugging and exception handling. Earlier simpler treatments of files, LINQ, databases and ASP.NET web app development. This book contains the core content for introductory courses. Twelve "optional" online chapters are included for second courses and professionals: WPF GUI and graphics, WPF multimedia, XML(R), collections, Silverlight(TM), web services, an OO design case study and more. New books purchased from Pearson include free, web-based student supplements on the Companion Website (see the Access Card at the front of the book). The Companion Website also includes VideoNotes for most of the code examples in the core chapters. For more information visit www.pearsonhighered.com/deitel Follow Deitel on Twitter(R) @deitel and Facebook(R) www.deitel.com/deitelfan/ Visit this book's website: www.deitel.com/books/vb2010http/ Contact the authors at: deitel@deitel.com Register for the DEITEL(R) BUZZ ONLINE newsletter at www.deitel.com/newsletter/subscribe.html Visit the Deitels' Visual Basic and other Microsoft-related Resource Centers at www.deitel.com/resourcecenters.html

The author introduces the reader to the creation and implementation of space-related models by applying a learning-by-doing and problem-oriented approach. The required procedural skills are rarely taught at universities and many scientists and engineers struggle to transfer a model into a computer program. The purpose of this book is to fill this gap. It moves from simple to more complex applications, covering various important topics in the sequence: dynamic matrix processing, 2D and 3D graphics, databases, Java applets and parallel computing. A file (SMOP.zip) with all examples can be downloaded free of charge from the Internet at <http://de.geocities.com/bsttc2/book>.

In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will build a project so that children can practice basic skills in addition, subtraction, multiplication, and division operations. This Math Game project can be used to choose the types of questions and what factors you want to use. This project has three timing options. Random math problems using values ??from 0 to 9 will be presented. Timing options are provided to measure accuracy and speed. There are many controls used. Two label controls are used for title information, two for displaying scores. There is a wide label in the middle of the form to display math questions. And, long skinny label is used as separator. Two button controls are used to start and stop question and one button to exit the project. There are three group control boxes. The first group box holds four check box controls that are used to select the type of questions. The second group box holds eleven radio buttons that are used to select values ??that are used as factors in calculations. The third group box contains three radio button controls for timing options. A scroll bar control rod is used to change the time. In chapter three, you will build Bank Code game. The storage box is locked and can only be opened if you enter the correct digit combination. Combinations can be 2 to 4 non-repetitive digits (range of digits from 1 to 9). After a guess is given, you will be notified of how many digits are right and how many digits are in the right position. Based on this information, you will give another guess. You continue to guess until you get the right combination or until you stop the game. On the left side of the form is a large picture box control. On the right side, two group box controls and two button controls are placed. In the picture box, a control panel is placed. In the panel, there are four label controls (set the AutoSize property to False) and nine button controls. In the first group box control, place three radio buttons. In the second group box control, a text box control is placed. The picture box contains an image of bank and a panel. The label controls in the panel are used to display the combinations entered (the BorderStyle property set to FixedSingle to display the label size). The nine buttons on the panel are used to enter combinations. Radio buttons are used to set options. The buttons (one to start and stop the game and another to exit the project) are used to control game operations. The text box displays the results of the combinations entered. In chapter four, you will build Horse Racing game. This is a simple game. Up to 10 horses will race to the finish line. You guessed two horses that you thought could win the race. By clicking on the Start button, the race will start. All horses will race speed to get to the finish line. Labels are used to display instructions and number of horses in a race. Four button controls are used: two buttons to change number of horses, one button to start the game, and one other button to stop the game. The picture box control is used to load the horse image. A timer control is used to update the horse's movement during the race. In chapter five, you will build Catching Ball game. The bird flew and dropped ball from the sky. Users are challenged to position man under the fallen ball to catch it. Labels are used for instructions and to display game information (remaining time, number of balls captured, and game difficulty level). Two buttons are used to change the game difficulty level, one button to start the game, and another button to stop the game. Picture box controls hold images for man, bird, and ball. In chapter six, you will build Smart Tic Tac Toe game. That said, this is the first game ever programmed on a computer and one that had been programmed by Bill Gates himself when he was a teenager while attending Lakeside School in Seattle. The aim of this game is to win the game on a 3 x 3 grid with the victory of three identical symbols (X or O) on horizontal, diagonal, or vertical lines. The players will play alternately. In this game given two game options: player 1 against player 2 or human player against computer. A smart but simple strategy will be developed for computer logic to be a formidable opponent for humans. In chapter seven, you will build Fighting Plane program. This program can be played by two human players or human player versus computer. The controls of the player are done via the keyboard. Player 1 presses A key to move up, Z key to move down, and S key to throw rudal. When you choose Two players from the Options button, this game can be played by two human players. Player 1 presses the same keys, while player 2 presses key K to move up, M to move down, and key J to throw rudal. All label controls are used for titles and provide scoring and game information. The large panel (Panel1) is the playing field. Three button controls are used to start / stop a

program, set options, and exit the program. One timer control is used to control game animation and another is used to represent the computer's decision process. The second control panel (Panel2) is used to select game options. One group box contains radio buttons which are used to select number of players. A group box contains radio buttons to select the level of difficulty of the game, when playing against a computer. A small button is used to close the options panel. The default properties are set for one-player games with the easiest game difficulty.

Removes the scanning artefacts and transmission imperfections to produce a most comprehensive and beautifully detailed set of images of the lunar surface. To help practical astronomers, all the photographs are systematically related to an Earth-based view. Organized to make it easy for astronomers to use, enabling ground-based images and views to be compared with the Orbiter photographs.

This is an introduction to programming using Microsoft's Visual Basic.NET 2010, intended for novice programmers with little or no programming experience or no experience with Visual Basic. The text emphasizes programming logic and good programming techniques with generous explanations of programming concepts written from a non-technical point of view. It stresses input, processing, and output and sequence, selection, and repetition in code development. File I/O and arrays are included. Later chapters introduce objects, event programming, and databases. By taking a slow and steady approach to programming ideas, this book builds new concepts from what the reader has already learned. VB tips and quips inject both humor and insight. The book includes numerous programming examples and exercises, case studies, tutorials, and 'fixing a program' sections for an in-depth look at programming problems and tools. Quizzes and review questions throughout each chapter get students to think about the materials and how to use them. Each chapter has a summary and glossary for extra review. The accompanying website, www.cambridge.org/us/McKeown, has code downloads, I/O, and database files from small, simple files to large files with thousands of records, flowcharts, deskchecks and audits to aid with program design, coding, and debugging; PowerPoint files for every chapter; and hundreds of ideas for programs and projects.

Visual Basic Graphics Programming Hands-On Applications and Advanced Color Development John Wiley & Sons

BOOK 1: VISUAL BASIC .NET AND DATABASE: PRACTICAL TUTORIALS This book aims to develop a MySQL-driven desktop application that readers can develop for their own purposes to implement library project using Visual Basic .NET. In Tutorial 1, you will build a Visual Basic interface for the database. This interface will be used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. You will create login form. Place on the form one picture box, two labels, one combo box, one text box, and two buttons. In Tutorial 2, you will build a school inventory project where you can store information about valuables in school. The table will have nine fields: Item (description of the item), Quantity, Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). In Tutorial 3, you will perform the steps necessary to add 5 new tables using phpMyAdmin into Academy database. You will build each table and add the associated fields as needed. Every table in the database will need input form. In this tutorial, you will build such a form for Author table. Although this table is quite simple (only four fields: AuthorID, Name, BirthDate, and PhotoFile), it provides a basis for illustrating the many steps in interface design. SQL statement is required by the Command object to read fields (sorted by Name). Then, you will build an interface so that the user can maintain the Publisher table in the database (Academy). The Publisher table interface is more or less the same as Author table interface. This Publisher table interface only requires more input fields. So you will use the interface for the Author table and modify it for the Publisher table. In Tutorial 4, you will perform the steps necessary to design and implement title form, library member form, and book borrowing form. You start by designing and testing the basic entry form for book titles. The Title table has nine fields: BookTitle, PublishYear, ISBN, PublisherID, AuthorID, Description, Note, Subject, and Comment. Then, you will build such a form for Member table. This table has twelve fields: MemberID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, and PhotoFile). You need thirteen label controls, one picture box, six text boxes, four comboboxes, one check box, one date time picker, one openFileDialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching member's name, and one button to upload member's photo. Finally, you will build such a form for Borrow table. This table has seven fields: BorrowID, MemberID, BorrowCode, ISBN, BorrowDate, ReturnDate, and Penalty. In this form, you need fourteen label controls, seven text boxes, two comboboxes, two date time pickers, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for other utilities, one button to generate borrowing code, and one button to return book.

BOOK 2: LEARN FROM SCRATCH VISUAL BASIC .NET WITH MYSQL This book will teach you with step-by-step approach to develop from scratch a MySQL-driven desktop application that readers can develop for their own purposes to implement school database project using Visual Basic .NET. In Tutorial 1, you will perform the steps necessary to add 8 tables using phpMyAdmin into School database that you will create. You will build each table and add the associated fields as needed. In this tutorial, you will also build login form and main form. In Tutorial 2, you will build such a form for Parent table. This table has thirteen fields: ParentID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need fourteen label controls, two picture boxes, six text boxes, four comboboxes, one check box, one date time picker, one openFileDialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for other utilities, one button for searching member's name, one button to upload parent's photo, and button to upload parent's finger. Place these controls on the form. In Tutorial 3, you will build such a form for Student table. This table has fifteen fields: StudentID, ParentID, FirstName, LastName, BirthDate, YearEntry, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need sixteen label controls, two picture boxes, six text boxes, five comboboxes, one check box, two date time pickers, one openFileDialog, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for controlling editing features, one button for searching parent's name, one button to open parent form, one button to upload student's photo, and one button to upload student's finger. In Tutorial 4, you will build a form for Teacher table. This table has fifteen fields: TeacherID, RegNumber, FirstName, LastName, BirthDate, Rank, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and

FingerFile). You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need sixteen label controls, one picture box, seven text boxes, five comboboxes, one check box, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching teacher's name, and one button to upload teacher's photo. In Tutorial 5, you will build a form for Subject table. This table has only three fields: SubjectID, Name, and Description. You need four label controls, four text boxes, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for utilities, and one button for searching subject name. Place these controls on the form. You will also build a form for Grade table. This table has seven fields: GradeID, Name, SubjectID, TeacherID, SchoolYear, TimeStart, and TimeFinish. You need to add seven label controls, one text box, four comboboxes, and two date time pickers. You also need four buttons for navigation, seven buttons for controlling editing features, one button to open subject form, and one button to open teacher form. In Tutorial 6, you will build a form for Grade_Student table. This table has only three fields: Grade_StudentID, GradeID, and StudentID. You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need two label controls and two comboboxes. You also need four buttons for navigation, seven buttons for controlling editing features, one button to open grade form, and one button to open student form.

This book is a comprehensive guide to Python as one of the fastest-growing computer languages including Web and Internet applications. This clear and concise introduction to the Python language is aimed at readers who are already familiar with programming in at least one language. This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of PostgreSQL and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to PostgreSQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will get introduction of postgresql. And then, you will learn querying data from the postgresql using Python including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using Python, updating data in postgresql database using Python, calling postgresql stored function using Python, deleting data from a postgresql table using Python, and postgresql Python transaction. In chapter seven, you will create and configure PostgreSQL database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have a VARCHAR data type (200). You will also create GUI to display, edit, insert, and delete for this table. In chapter nine, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In chapter ten, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

In Tutorial 1, you will start building a Visual C# interface for database management system project using MySQL. The database, named DBMS, is created. The designed interface in this tutorial will be used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. In Tutorial 2, you will build a project, as part of database management system, where you can store information about valuables in school. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). In Tutorial 3 up to Tutorial 4, you will perform the steps necessary to add 6 tables using phpMyAdmin into DBMS database. You will build each table and add the associated fields as needed. In this tutorial, you will create a library database project, as part of database management system, where you can store all information about library including author, title, and publisher. In Tutorial 5 up to Tutorial 7, you will perform the steps necessary to add 8 more tables using phpMyAdmin into DBMS database. You will build each table and add the associated fields as needed. In this tutorial, you will create a high school database project, as part of database management system, where you can store all information about school including parent, teacher, student, subject, and, title, and grade.

In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying

them to some applications in this book. In chapter two, you will go through step by step to build a SALES database using Microsoft Access and SQL Server. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the client ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. If the small lock symbol is not displayed next to the ClientID row, then you need to place it there. Right click on ClientID row and select Primary Key. A small key is now displayed next to the entry indicating it is the primary key. You will define FamilyName as an index. Select the FamilyName line. On the General tab, set the Indexed property to Yes (Duplicates OK). You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. In chapter three, you will build a Visual C# interface for the database. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will have the ability to add new orders, find clients, add new clients. The completed order invoice will be provided in a printed report. In chapter four, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter five, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. You will do this in stages, from database development to creation of distribution packages. These steps are the same as those used in developing a commercial database application. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter six, you will perform the steps necessary to build a SQL Server book inventory database that contains 4 tables using Microsoft Visual Studio 2019. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table.

Visual informatics is a field of interest not just among the information technology and computer science community, but also other related fields such as engineering, medical and health informatics and education starting in the early 1990s. Recently, the field is gaining more attention from researchers and industry. It has become a multidisciplinary and trans-disciplinary field related to research areas such as computer vision, visualization, information visualization, real-time image processing, medical image processing, image information retrieval, virtual reality, augmented reality, -pressive visual mathematics, 3D graphics, multimedia-fusion, visual data mining, visual ontology, as well as services and visual culture. Various efforts have been invested in different research, but operationally, many of these systems are not prominent in the mass market and thus knowledge and research on these phenomena within the mentioned areas need to be shared and disseminated. It is for this reason that the Visual Informatics Research Group from Universiti - bangsaan Malaysia (UKM) decided to spearhead this initiative to bring together experts in this very diversified but important research area so that more concerted efforts can be undertaken not just within the visual informatics community in Malaysia but from other parts of the world, namely, Asia, Europe, Oceania, and USA. This first International Visual Informatics Conference (IVIC 2009) was conducted collaboratively, by the visual informatics research community from the various public and private institutions of higher learning in Malaysia, and hosted by UKM.

PCMag.com is a leading authority on technology, delivering Labs-based, independent reviews of the latest products and services. Our expert industry analysis and practical solutions help you make better buying decisions and get more from technology.

This hands-on introduction to database programming using Java is ideal for people with little or no programming experience. The goal of this concise book is not just to teach you Java, but to help you think like a programmer. Each brief chapter covers the material for one week of a college course to help you practice what you've learned. As you would expect, this book shows how to build from scratch two different databases: MySQL and SQLite using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In chapter four, you will study how to query the six tables. In chapter five, you will be shown how to create SQLite database and tables with Java. In chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. Digital image techniques to extract image features used in this chapter are grascaling, sharpening, inverting, blurring, dilation, erosion, closing, opening, vertical prewitt, horizontal prewitt, Laplacian, horizontal sobel, and vertical sobel. For readers, you can develop it to store other advanced image features based on descriptors such as SIFT and others for developing descriptor based

matching. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. All six fields (except keys) will have a BLOB data type, so that the image of the feature will be directly saved into this table. In chapter nine, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and Case_File. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File has seven columns: case_file_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MySQL/SQLite programmer.

This hands-on book introduces the essential topic of coding and the Python computer language to beginners and programmers of all ages. This book explains relational theory in practice, and demonstrates through two projects how you can apply it to your use of MySQL and SQL Server databases. This book covers the important requirements of teaching databases with a practical and progressive perspective. This book offers the straightforward, practical answers you need to help you do your job. This hands-on tutorial/reference/guide to MySQL and SQL Server is not only perfect for students and beginners, but it also works for experienced developers who aren't getting the most from both databases. In designing a GUI and as an IDE, you will make use Qt Designer. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create and configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables.

In this book, you will learn how to build from scratch a MySQL database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use MySQL in Java. In the first chapter, you will learn: How to install NetBeans, JDK 11, and MySQL Connector/J; How to integrate external libraries into projects; How the basic MySQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the third chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In the last chapter, you will study how to query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/MySQL programmer.

Combining the Deitel™ signature Live-Code™ Approach with a new Application-Driven™ methodology, this text uses a step-by-step tutorial approach to begin teaching students the basics of programming, builds upon previously learned concepts, and introduces new programming features in each successive tutorial. KEY TOPICS This comprehensive introduction to Visual Basic .NET covers GUI design, controls, methods, functions, data types, control structures, procedures, arrays, object-oriented programming, strings and characters, sequential files, and more. It also includes higher-end topics such as database programming, multimedia and graphics, and Web applications development. For individuals beginning their mastery of Visual Basic Programming.

Book 1: VISUAL C# .NET WITH MYSQL: A Definitive Guide to Develop Database-Oriented Desktop Applications In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will go through step by step to build a SALES

database using MySQL. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the client ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. You will define FamilyName as an index. You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will have the ability to add new orders, find clients, add new clients. The completed order invoice will be provided in a printed report. In chapter three, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter four, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter five, you will perform the steps necessary to build a MySQL book inventory database that contains 4 tables. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table. Book 2: Visual C# .NET For Programmers: A Progressive Tutorial to Develop Desktop Applications In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will go through step by step to build a SALES database using Microsoft Access and SQL Server. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the client ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. If the small lock symbol is not displayed next to the ClientID row, then you need to place it there. Right click on ClientID row and select Primary Key. A small key is now displayed next to the entry indicating it is the primary key. You will define FamilyName as an index. Select the FamilyName line. On the General tab, set the Indexed property to Yes (Duplicates OK). You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. In chapter three, you will build a Visual C# interface for the database. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will have the ability to add new orders, find clients, add new clients. The completed order invoice will be provided in a printed report. In chapter four, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter five, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. You will do this in stages, from database development to creation of distribution packages. These steps are the same as those used in developing a commercial database application. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter six, you will perform the steps necessary to build a SQL Server book inventory database that contains 4 tables using Microsoft Visual Studio 2019. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table. In this book, you will learn how to build from scratch a PostgreSQL database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to utilize PostgreSQL in Java. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the first chapter, you will learn: How to install NetBeans, JDK 11, and the PostgreSQL connector; How to integrate external libraries into projects; How the basic PostgreSQL commands are used; How to query statements to create databases, create tables, fill tables, and manipulate table contents is done. In the second chapter, you will learn querying data from the postgresql using jdbc including establishing a database connection, creating a statement object, executing the query, processing the resultset object, querying data using a statement that returns multiple rows, querying data using a statement that has parameters, inserting data into a table using jdbc, updating data in postgresql database using jdbc, calling postgresql stored function using jdbc, deleting data from a postgresql table using jdbc, and postgresql jdbc transaction. In the third chapter, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In the fourth chapter, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In the last chapter, you will study how to

query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/PostgreSQL programmer.

In chapter one, you will get to know the properties and events of each control in a Windows Visual Basic application. You need to learn and know in order to be more familiar when applying them to some desktop applications in this book. In Tutorial 1.1, you will build a dual-mode stopwatch. The stopwatch can be started and stopped whenever desired. Two time traces: the running time when the stopwatch is active (running time) and the total time since the first stopwatch was activated. Two label controls are used to display the time (two more labels to display title information). Two button controls are used to start/stop and reset the application, one more button to exit the application. The timer control is used to periodically (every second) update the displayed time. In Tutorial 1.2, you will build a project so that children can practice basic skills in addition, subtraction, multiplication, and division operations. This Math Game project can be used to choose the types of questions and what factor you want to use. This project has three timing options. In Tutorial 1.3, you will build Bank Code game. The storage box is locked and can only be opened if you enter the correct digit combination. Combinations can be 2 to 4 non-repetitive digits (range of digits from 1 to 9). After a guess is given, you will be notified of how many digits are right and how many digits are in the right position. Based on this information, you will give another guess. You continue to guess until you get the right combination or until you stop the game. In Tutorial 1.4, you will build Horse Racing game. This is a simple game. Up to 10 horses will race to the finish line. You guessed two horses that you thought could win the race. By clicking on the Start button, the race will start. All horses will race speed to get to the finish line. In chapter two, you will learn the basic concepts of classes and objects. Next, it will demonstrate how to define class and type of enumeration, which shows how both are used in the application. In Tutorial 2.1, you will create a two-level application that uses a form to pass input user to the People class. The form class is the level of representation and the People class is the middle level. You will add controls to the form so people can enter ID, last name, and their height. When the user clicks the Save button, the code will assign input values ??to the People class properties. Finally, you will display the People object on a label. Figure below shows the form after the user clicks the Save button. In Tutorial 2.2, you will add a parameterized constructor to the People class. The application will ask the user to enter values, which will then be passed to the People constructor. Then, the application will display the values ??stored on the People object. In Tutorial 2.3, you will create an application that utilizes enumeration type. The user will choose one type of account that is listed in a ListBox control and what he chooses is then displayed in a Label control. In Tutorial 2.4, you will create a simple Bank application. This application has one class, BankAcc, and a startup form. In Tutorial 2.5, you will improve the simple Bank application, by implementing the following two properties in the BankAcc class: TotalDeposit- Total money saved in current account; TotalWithdraw- Total funds that have been withdrawn from current account. In Tutorial 2.6, you will create an application to calculate the time needed for a particular aircraft to reach takeoff speed. You will also calculate how long the runway will be required. For each type of aircraft, you are given (1) the name of the aircraft, (2) the required take-off speed (feet/sec), and (3) how fast the plane accelerates (feet/sec²). In Tutorial 2.7, you will provide a number of programming training for those who want to improve their programming skills. Your task here is to write an object-oriented application so that training manager can display and edit the training services offered. There are several training categories: (1) Application Development, (2) Database, (3) Networking, and (4) System Administration. The training itself consists of: (1) title, (2) training days, (3) category, and (4) cost. Create a class named Training that contains this information, along with its properties and a ToString() method. In chapter three, several tutorials will be presented to build more complex projects. You will build them gradually and step by step. In Tutorial 3.1, you will build Catching Ball game. The bird flew and dropped ball from the sky. User is challenged to position man under the fallen ball to catch it. In Tutorial 3.2, you will build Smart Tic Tac Toe game. The aim of this game is to win the game on a 3 x 3 grid with the victory of three identical symbols (X or O) on horizontal, diagonal, or vertical lines. The players will play alternately. In this game given two game options: player 1 against player 2 or human player against computer. A smart but simple strategy will be developed for computer logic to be a formidable opponent for human. In Tutorial 3.3, you will build a Matching Images game. Ten pairs of images hidden on the game board. The object of the game is to find image pairs. In Two Players mode, players will get turns in turn. In One Player mode, there are two options to choose from: Playing Alone or Against Computer. When Play Alone option is selected, the player will play alone without an opponent. If Against Computer option is selected, then the level of computer intelligence is given with several levels according to the level of difficulty of the game. In Tutorial 3.4, you will build Throwing Fire program. This program can be played by two human players or human player versus computer. In chapter four, tutorials will be presented to build two advanced projects. You will build them gradually and step by step. In Tutorial 4.1, you will build Roasted Duck Delivery simulation. In this simulation, a number of decisions are needed. The basic idea is to read the order by incoming telephone and tell the delivery scooter to go to the location of the order. You also need to make sure that you always provide a roasted duck ready to be transported by the delivery scooter. The delivery area is a 20 by 20 square grid. The more roasted duck is sold, the more profit it gets. In Tutorial 4.2, you will build a Drone Simulation. In this simulation, you control both vertical and horizontal thrusters to maneuver the ride to the landing pad. You will adjust the landing speed so that it is slow enough so that no accident occurs.

An arsenal of more than 170 proven techniques and ready-to-run source code for better and more efficient programming Ready-to-Run Visual Basic(r) Code Library Thousands of VB programmers the world over have benefited from the stockpile of ready-to-run programming techniques available on Rod Stephens' VB Helper Web site. Now, by popular demand for more working programs, here are 173 never-before-published basic, intermediate, and advanced techniques that will help you successfully tackle real-world programming situations you'll encounter. Grouped by topic for easy reference, the ready-to-run programs contained in this book/CD package provide solutions for many aspects of VB programming, including: * Manipulating numbers and working with variables * Manipulating databases * Handling text and formatting output * Getting the most out of ListBox, ComboBox, ListView, TreeView, ProgressBar, TabStrip, Multimedia MCI, and other VB controls * Creating menus * Handling forms * Animation * Networking The CD-ROM provides you with: * Ready-to-run source code for all the examples in the book Visit the author's comprehensive Visual Basic site at: www.vb-helper.com Wiley Computer Publishing Timely. Practical. Reliable. Visit our Web site at www.wiley.com/compbooks/

You will learn Java/MySQL fast, easy and fun. This book provides you with a complete MySQL guidance presented in an easy-to-follow manner. Each chapter has practical examples with SQL script and screenshots available. If you go through the entire chapters, you will know how to manage MySQL databases and manipulate data using various techniques such as MySQL queries, MySQL stored procedures, database views, triggers. In the first part of the book, you will learn Basic MySQL statements including how to implement querying data, sorting data, filtering data, joining tables, grouping data, subquerying data, dan setting operators. Aside from learning basic SQL statements, you will also learn step by step how to develop stored procedures in MySQL. First, we introduce you to the stored procedure concept and discuss when you should use it. Then, we show you how to use the basic elements of the procedure code such as create procedure statement, if-else, case, loop, stored procedure's parameters. In the next chapter, we will discuss the database views, how they are implemented in MySQL, and how to use them more effectively. After that, you will learn how to work with the MySQL triggers. By definition, a trigger or database trigger is a stored program executed automatically to respond to a specific event e.g., insert, update or delete occurred in a table. The database trigger is powerful tool for protecting the integrity of the data in your MySQL databases. In addition, it is useful to automate some database operations such as logging, auditing, etc. Then, you will learn about MySQL index including creating indexes, removing indexes, listing all indexes of a table and other important features of indexes in MySQL. MySQL uses indexes to quickly find rows with specific column values. Without an index, MySQL must scan the whole table to locate the relevant rows. The larger table, the slower it searches. After that, you will find a lot of useful MySQL administration techniques including MySQL server

startup and shutdown, MySQL server security, MySQL database maintenance, and backup. The last chapter gives you the most commonly used MySQL functions including aggregate functions, string functions, date time functions, control flow functions, etc.

This Wrox Blox shows you how to add graphics to Visual Basic 2008 applications by explaining fundamental graphics techniques such as: drawing shapes with different colors and line styles; filling areas with colors, gradients, and patterns; drawing text that is properly aligned, sized, and clipped exactly where you want it; manipulating images and saving results in bitmap, JPEG, and other types of files. Also covered are instructions for how to greatly increase your graphics capabilities using transformations, which allow you to move, stretch, or rotate graphics. They also let you work in coordinate systems that make sense for your application. The author also describes techniques for using the above in printouts, describing the sequence of events that produce a printout and show how to generate and preview printouts, with examples which show how to wrap long chunks of text across multiple pages, if necessary. In addition, you will learn about two powerful new graphic tools that were introduced with .NET Framework 3.0: WPF graphics and FlowDocuments. XAML graphic commands allow a WPF application to draw and fill the same kinds of shapes that a program can draw by using graphics objects. Finally, a discussion on the FlowDocument object shows you how to define items that should be flowed across multiple pages as space permits. This lets you display text, graphics, controls, and other items that automatically flow across page breaks. FlowDocument viewers make displaying these documents easy for you, and simplifies the user's reading of the documents. This Wrox Blox also contains 35 example programs written in Visual Basic 2008, although most of the code works in previous versions of Visual Basic .NET as well. The most notable exceptions are WPF graphics and FlowDocuments, both of which require WPF provided in .NET Framework 3.0 and later.

For programmers who don't know Visual Basic or have CCE, this book allows them to take advantage of this technology, providing over 101 prebuilt custom controls that can be plugged in to any of the nine major programming languages. CD-ROM includes 101 prebuilt, ready-to-run custom controls ready to plug into any program created in one of the nine basic languages, source code, test, and Visual Basic 5 Control Creation Edition.

This book aims to develop a database-driven desktop application that readers can develop for their own purposes to implement database-oriented digital image processing, machine learning, and image retrieval applications. In Tutorial 1, you will perform the steps necessary to add 6 tables using Visual C# into ImageProc database. You will build each table and add the associated fields as needed. In this tutorial, you will also build such a form for Officer table. This table has sixteen fields: OfficerID, FirstName, LastName, RegNumber, BirthDate, AppDate, Gender, Status, Rank, Address, Mobile, Phone, Email, Description, PhotoFile, and FingerFile). You need seventeen label controls, two picture boxes, ten text boxes, two comboboxes, one check box, two date time pickers, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, eight buttons for utilites, one button for searching officer's name, one button to upload officer's photo, and one button to upload officer's fingerprint. In Tutorial 2, you will perform the steps necessary to create and implement police station form. In this tutorial, you will build such a form for PoliceStation table. This table has seven fields: PSID, OfficerID, PSName, City, Address, Phone, and Description. You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need eight label controls, six text boxes, two comboboxes, one check box, and one printpreviewdialog. You also need four buttons for navigation, eight buttons for utilites, and one button for searching officer's name. Place these controls on the form. In Tutorial 3, you will build such a form for Accused table. This table has thirteen fields: AccusedID, FullName, MotherName, CrimeCase, BirthDate, Gender, Address, Mobile, Phone, Email, Description, PhotoFile, and FingerFile). You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need fourteen label controls, two picture boxes, nine text boxes, two comboboxes, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, eight buttons for utilites, one button for searching accused's name, one button to upload accused's photo, and one button to upload accused's fingerprint. In Tutorial 4, you will build such a form for Witness table. This table has thirteen fields: WitnessID, FullName, MotherName, CrimeCase, BirthDate, Gender, Address, Mobile, Phone, Email, Description, PhotoFile, and FingerFile). You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need fourteen label controls, two picture boxes, nine text boxes, two comboboxes, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, eight buttons for utilites, one button for searching witness's name, one button to upload witness's photo, and one button to upload witness's fingerprint. In Tutorial 5, you will build such a form for Victim table. This table has thirteen fields: VictimID, FullName, MotherName, CrimeCase, BirthDate, Gender, Address, Mobile, Phone, Email, Description, PhotoFile, and FingerFile). You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need fourteen label controls, two picture boxes, nine text boxes, two comboboxes, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, eight buttons for utilites, one button for searching victim's name, one button to upload victim's photo, and one button to upload victim's fingerprint. In Tutorial 6, you will build such a form for CrimeReg table. This table has fourteen fields: CRID, CRNumber, PSID, VictimID, AccusedID, DateReport, DateCrime, Arrested, CaseStatus, Description, Feature1, Feature2, Feature3, and Feature4. You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need thirty two label controls, seven text boxes, ten comboboxes, one check box, two date time pickers, six picture boxes, and one printpreviewdialog. You then need four buttons for navigation, eight buttons for utilites, and one button for searching crime register number. You also need button to save every feature.

In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will go through step by step to build a SALES database using MySQL. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the clien ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. You will define FamilyName as an index. You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will

have the ability to add new orders, find clients, add new clients. The completed order invoice will be provided in a printed report. In chapter three, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter four, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter five, you will perform the steps necessary to build a MySQL book inventory database that contains 4 tables. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table.

Book 1: This book aims to develop a MySQL-driven desktop application that readers can develop for their own purposes to implement library project using Visual Basic .NET. In Tutorial 1, you will build a Visual Basic interface for the database. This interface will used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. You will create login form. Place on the form one picture box, two labels, one combo box, one text box, and two buttons. In Tutorial 2, you will build a school inventory project where you can store information about valuables in school. The table will have nine fields: Item (description of the item), Quantity, Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). In Tutorial 3, you will perform the steps necessary to add 5 new tables using phpMyAdmin into Academy database. You will build each table and add the associated fields as needed. Every table in the database will need input form. In this tutorial, you will build such a form for Author table. Although this table is quite simple (only four fields: AuthorID, Name, BirthDate, and PhotoFile), it provides a basis for illustrating the many steps in interface design. SQL statement is required by the Command object to read fields (sorted by Name). Then, you will build an interface so that the user can maintain the Publisher table in the database (Academy). The Publisher table interface is more or less the same as Author table interface. This Publisher table interface only requires more input fields. So you will use the interface for the Author table and modify it for the Publisher table. Book 2: In chapter one, you will learn to know the properties and events of each control in a Windows Visual C# application. You need to learn and know in order to be more familiar when applying them to some applications in this book. In chapter two, you will go through step by step to build a SALES database using MySQL. You will build each table and add associated data fields (along with the necessary keys and indexes). The first field in the Client table is ClientID. Enter the clien ID in the Name Field and select AutoNumber in the Data Type. You define primary key and other indexes which are useful for quick searching. ClientID is a primary field. You will define FamilyName as an index. You then will create Ordering table with three fields: OrderID, ClientID, and OrderDate. You then will create Purchase table with three fields: OrderID, ProductID, and Quantity. And you will create Product table with four fields: ProductID, Description, Price, and QtySold. Before designing Visual C# interface, you will build the relationships between four tables. The interface will be used to enter new orders into the database. The order form will be used to enter the following information into the database: order ID, order date, client ID, client's first name and family name, client's address, product information ordered. The form will have the ability to add new orders, find clients, add new clients. The completed order invoice will be provided in a printed report. In chapter three, you will build a database management system where you can store information about valuables in your warehouse. The table will have seven fields: Item (description of the item), Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). The development of this Warehouse Inventory Project will be performed, as usual, in a step-by-step manner. You will first create the database. Furthermore, the interface will be built so that the user can view, edit, add, or add data records from the database. Finally, you add code to create a printable list of information from the database. In chapter four, you will build an application that can be used to track daily high and low pollutant PM2.5 and air quality level. The steps that need to be taken in building Siantar Air Quality Index (SAQI) database project are: Build and test a Visual C# interface; Create an empty database using code; and Report database. The designed interface will allow the user to enter max pollutant, min pollutant, and air quality for any date that the user chooses in a particular year. This information will be stored in a database. Graphical result of the data will be provided, along with summary information relating to the maximum value, minimum value, and mean value. You will use a tab control as the main component of the interface. The control has three tabs: one for viewing and editing data, one for viewing graph of pollutant data, and another for viewing graph of air quality data. Each tab on this control operates like a Visual C# control panel. In chapter five, you will perform the steps necessary to build a MySQL book inventory database that contains 4 tables. You will build each table and add the associated fields as needed. You will have four tables in the database and define the relationship between the primary key and foreign key. You will associate AuthorID (foreign key) field in the Title_Author table with AuthorID (primary key) in the Author table. Then, you want to associate the ISBN (foreign key) field in Title_Author table with ISBN (primary key) in the Title table.

The contributions for this book have been gathered over several years from conferences held in the series of Mechatronics and Machine Vision in Practice, the latest of which was held in Ankara, Turkey. The essential aspect is that they concern practical applications rather than the derivation of mere theory, though simulations and visualization are important components. The topics range from mining, with its heavy engineering, to the delicate machining of holes in the human skull or robots for surgery on human flesh. Mobile robots continue to be a hot topic, both

from the need for navigation and for the task of stabilization of unmanned aerial vehicles. The swinging of a spray rig is damped, while machine vision is used for the control of heating in an asphalt-laying machine. Manipulators are featured, both for general tasks and in the form of grasping fingers. A robot arm is proposed for adding to the mobility scooter of the elderly. Can EEG signals be a means to control a robot? Can face recognition be achieved in varying illumination?"

This book will teach you with step-by-step approach to develop from scratch a MySQL-driven desktop application that readers can develop for their own purposes to implement school database project using Visual Basic .NET. In Tutorial 1, you will perform the steps necessary to add 8 tables using phpMyAdmin into School database that you will create. You will build each table and add the associated fields as needed. In this tutorial, you will also build login form and main form. In Tutorial 2, you will build such a form for Parent table. This table has thirteen fields: ParentID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need fourteen label controls, two picture boxes, six text boxes, four comboboxes, one check box, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for other utilities, one button for searching member's name, one button to upload parent's photo, and button to upload parent's finger. Place these controls on the form. In Tutorial 3, you will build such a form for Student table. This table has fifteen fields: StudentID, ParentID, FirstName, LastName, BirthDate, YearEntry, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need sixteen label controls, two picture boxes, six text boxes, five comboboxes, one check box, two date time pickers, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for controlling editing features, one button for searching parent's name, one button to open parent form, one button to upload student's photo, and one button to upload student's finger. In Tutorial 4, you will build a form for Teacher table. This table has fifteen fields: TeacherID, RegNumber, FirstName, LastName, BirthDate, Rank, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, PhotoFile, and FingerFile). You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need sixteen label controls, one picture box, seven text boxes, five comboboxes, one check box, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching teacher's name, and one button to upload teacher's photo. In Tutorial 5, you will build a form for Subject table. This table has only three fields: SubjectID, Name, and Description. You need four label controls, four text boxes, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for utilities, and one button for searching subject name. Place these controls on the form. You will also build a form for Grade table. This table has seven fields: GradeID, Name, SubjectID, TeacherID, SchoolYear, TimeStart, and TimeFinish. You need to add seven label controls, one text box, four comboboxes, and two date time pickers. You also need four buttons for navigation, seven buttons for controlling editing features, one button to open subject form, and one button to open teacher form. In Tutorial 6, you will build a form for Grade_Student table. This table has only three fields: Grade_StudentID, GradeID, and StudentID. You need an input form so that user can edit existing records, delete records, or add new records. The form will also have the capability of navigating from one record to another. You need two label controls and two comboboxes. You also need four buttons for navigation, seven buttons for controlling editing features, one button to open grade form, and one button to open student form. This book aims to develop a MySQL-driven desktop application that readers can develop for their own purposes to implement library project using Visual Basic .NET. In Tutorial 1, you will build a Visual Basic interface for the database. This interface will be used as the main terminal in accessing other forms. This tutorial will also discuss how to create login form and login table. You will create login form. Place on the form one picture box, two labels, one combo box, one text box, and two buttons. In Tutorial 2, you will build a school inventory project where you can store information about valuables in school. The table will have nine fields: Item (description of the item), Quantity, Location (where the item was placed), Shop (where the item was purchased), DatePurchased (when the item was purchased), Cost (how much the item cost), SerialNumber (serial number of the item), PhotoFile (path of the photo file of the item), and Fragile (indicates whether a particular item is fragile or not). In Tutorial 3, you will perform the steps necessary to add 5 new tables using phpMyAdmin into Academy database. You will build each table and add the associated fields as needed. Every table in the database will need input form. In this tutorial, you will build such a form for Author table. Although this table is quite simple (only four fields: AuthorID, Name, BirthDate, and PhotoFile), it provides a basis for illustrating the many steps in interface design. SQL statement is required by the Command object to read fields (sorted by Name). Then, you will build an interface so that the user can maintain the Publisher table in the database (Academy). The Publisher table interface is more or less the same as Author table interface. This Publisher table interface only requires more input fields. So you will use the interface for the Author table and modify it for the Publisher table. In Tutorial 4, you will perform the steps necessary to design and implement title form, library member form, and book borrowal form. You start by designing and testing the basic entry form for book titles. The Title table has nine fields: BookTitle, PublishYear, ISBN, PublisherID, AuthorID, Description, Note, Subject, and Comment. Then, you will build such a form for Member table. This table has twelve fields: MemberID, FirstName, LastName, BirthDate, Status, Ethnicity, Nationality, Mobile, Phone, Religion, Gender, and PhotoFile). You need thirteen label controls, one picture box, six text boxes, four comboboxes, one check box, one date time picker, one openfiledialog, and one printpreviewdialog. You also need four buttons for navigation, six buttons for controlling editing features, one button for searching member's name, and one button to upload member's photo. Finally, you will build such a form for Borrow table. This table has seven fields: BorrowID, MemberID, BorrowCode, ISBN, BorrowDate, ReturnDate, and Penalty. In this form, you need fourteen label controls, seven text boxes, two comboboxes, two date time pickers, and one printpreviewdialog. You also need four buttons for navigation, seven buttons for other utilities, one button to generate borrowal code, and one button to return book.

Prevent program bugs right from the start and quickly exterminate existing ones! Bug Proofing VISUAL BASIC(r) Let's face it, most programs with more than ten lines of code contain bugs. And as you know, the longer a bug exists in a system, the harder it becomes to locate and repair. Help is here! Packed with code, this practical guide shows you how to write effective, error-free programs and, best of all, how to test your programs at crucial stages of development. You'll find out how to handle unexpected bugs that do occur and how to locate and fix them quickly. You'll get bug prevention and repair techniques that just aren't available anywhere else! You'll get the kind of advice and information that usually only comes from years and years of hard-won experience! You'll find out: * How to design bug-free programs * How to code proactively to prevent bugs before they start * How to write code that exposes bugs instead of hiding them * How to catch bugs before they do any serious harm * How to find bugs using tools like the debugger and code profiler * How to use debug and runtime versions of a program to make debugging

easier * How to use On Error statements to handle unexpected conditions * How to record information automatically, so you can fix bugs after users encounter them * How to use proven methods to find errors quickly * How to create and analyze special foolproof tests for errors

This is a comprehensive, in-depth introduction to the core Java language book. This book will help you quickly write efficient, high-quality SQL-Server-based code with Java. It's an ideal way to begin, whether you're new to programming or a professional developer versed in other languages. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch a SQL Server database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQL Server in Java. In chapter one, you will be taught how to create Crime database and its tables. In chapter two, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter three, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter four, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter five, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter six, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables. In chapter seven, you will create School database and six tables. In chapter eight, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter nine, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six. In the last chapter, you will study how to query the six tables. Finally, this book is hopefully useful and can improve database programming skills for every Java/SQL Server programmer.

A world list of books in the English language.

The sample programs in this book were developed using Visual Basic 6. However, they can be easily modified to build applications for VB.Net. Visual Basic 6 is a third-generation event-driven programming language first released by Microsoft in 1991. In Visual Basic 6, the sky's the limit. You can develop all kinds of applications, including educational apps, financial apps, games, multimedia apps, animations, database applications and more. Visual Basic 6 Samples Code comprises 290 pages of captivating content and 48 fascinating sample codes. All the examples are explained in great detail using easy-to-understand language and illustrated with gorgeous Images. By reading the book and using the sample source codes, you will master Visual Basic programming effortlessly! You will be able to: - Understand basic to intermediate concepts of Visual Basic programming. - Create your own Visual Basic 6 programs from scratch. - Get programming ideas from 48 interesting sample programs. - Modify the source codes easily to suit your needs.

This covers how to implement SQLite and SQL Server driven Java GUI programming. The lessons in this book are a highly organized and well-indexed set of tutorials meant for students and programmers. Netbeans, a specific IDE (Integrated Development Environment) is used to create GUI (Graphical User Interface applications). The finished product is the reward, but the readers are fully engaged and enriched by the process. This kind of learning is often the focus of training. In this book, you will learn how to build from scratch a SQLite database management system using Java. In designing a GUI and as an IDE, you will make use of the NetBeans tool. Gradually and step by step, you will be taught how to use SQLite and SQL Server in Java. In chapter one, you will learn: How to create SQLite database and six tables In chapter two, you will study: Creating the initial three table projects in the school database: Teacher table, TClass table, and Subject table; Creating database configuration files; Creating a Java GUI for viewing and navigating the contents of each table; Creating a Java GUI for inserting and editing tables; and Creating a Java GUI to join and query the three tables. In chapter three, you will learn: Creating the main form to connect all forms; Creating a project will add three more tables to the school database: the Student table, the Parent table, and Tuition table; Creating a Java GUI to view and navigate the contents of each table; Creating a Java GUI for editing, inserting, and deleting records in each table; Creating a Java GUI to join and query the three tables and all six tables. In chapter four, you will study how to query the six tables. In chapter five, you will be taught how to create SQL Server database and its tables. In

chapter six, you will be taught how to extract image features, utilizing BufferedImage class, in Java GUI. In chapter seven, you will be taught to create Java GUI to view, edit, insert, and delete Suspect table data. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. In chapter eight, you will be taught to create Java GUI to view, edit, insert, and delete Feature_Extraction table data. This table has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. In chapter nine, you will add two tables: Police_Station and Investigator. These two tables will later be joined to Suspect table through another table, File_Case, which will be built in the seventh chapter. The Police_Station has six columns: police_station_id (primary key), location, city, province, telephone, and photo. The Investigator has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. Here, you will design a Java GUI to display, edit, fill, and delete data in both tables. In chapter ten, you will add two tables: Victim and File_Case. The File_Case table will connect four other tables: Suspect, Police_Station, Investigator and Victim. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The File_Case has seven columns: file_case_id (primary key), suspect_id (foreign key), police_station_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. Here, you will also design a Java GUI to display, edit, fill, and delete data in both tables.

Book 1: Practical Data Science Programming for Medical Datasets Analysis and Prediction with Python GUI In this book, you will implement two data science projects using Scikit-Learn, Scipy, and other libraries with Python GUI. In Project 1, you will learn how to use Scikit-Learn, NumPy, Pandas, Seaborn, and other libraries to perform how to predict early stage diabetes using Early Stage Diabetes Risk Prediction Dataset provided by Kaggle. This dataset contains the sign and symptom data of newly diabetic or would be diabetic patient. This has been collected using direct questionnaires from the patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh and approved by a doctor. You will develop a GUI using PyQt5 to plot distribution of features, feature importance, cross validation score, and predicted values versus true values. The machine learning models used in this project are Adaboost, Random Forest, Gradient Boosting, Logistic Regression, and Support Vector Machine. In Project 2, you will learn how to use Scikit-Learn, NumPy, Pandas, and other libraries to perform how to analyze and predict breast cancer using Breast Cancer Prediction Dataset provided by Kaggle. Worldwide, breast cancer is the most common type of cancer in women and the second highest in terms of mortality rates. Diagnosis of breast cancer is performed when an abnormal lump is found (from self-examination or x-ray) or a tiny speck of calcium is seen (on an x-ray). After a suspicious lump is found, the doctor will conduct a diagnosis to determine whether it is cancerous and, if so, whether it has spread to other parts of the body. This breast cancer dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. You will develop a GUI using PyQt5 to plot distribution of features, pairwise relationship, test scores, predicted values versus true values, confusion matrix, and decision boundary. The machine learning models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, and Support Vector Machine.

Book 2: Step by Step Tutorials For Data Science With Python GUI: Traffic And Heart Attack Analysis And Prediction In this book, you will implement two data science projects using Scikit-Learn, Scipy, and other libraries with Python GUI. In Chapter 1, you will learn how to use Scikit-Learn, Scipy, and other libraries to perform how to predict traffic (number of vehicles) in four different junctions using Traffic Prediction Dataset provided by Kaggle. This dataset contains 48.1k (48120) observations of the number of vehicles each hour in four different junctions: 1) DateTime; 2) Junction; 3) Vehicles; and 4) ID. In Chapter 2, you will learn how to use Scikit-Learn, NumPy, Pandas, and other libraries to perform how to analyze and predict heart attack using Heart Attack Analysis & Prediction Dataset provided by Kaggle.

Book 3: BRAIN TUMOR: Analysis, Classification, and Detection Using Machine Learning and Deep Learning with Python GUI In this project, you will learn how to use Scikit-Learn, TensorFlow, Keras, NumPy, Pandas, Seaborn, and other libraries to implement brain tumor classification and detection with machine learning using Brain Tumor dataset provided by Kaggle. This dataset contains five first order features: Mean (the contribution of individual pixel intensity for the entire image), Variance (used to find how each pixel varies from the neighboring pixel 0, Standard Deviation (the deviation of measured Values or the data from its mean), Skewness (measures of symmetry), and Kurtosis (describes the peak of e.g. a frequency distribution). It also contains eight second order features: Contrast, Energy, ASM (Angular second moment), Entropy, Homogeneity, Dissimilarity, Correlation, and Coarseness. The machine learning models used in this project are K-Nearest Neighbor, Random Forest, Naive Bayes, Logistic Regression, Decision Tree, and Support Vector Machine. The deep learning models used in this project are MobileNet and ResNet50. In this project, you will develop a GUI using PyQt5 to plot boundary decision, ROC, distribution of features, feature importance, cross validation score, and predicted values versus true values, confusion matrix, training loss, and training accuracy.

This book covers how to implement SQLite and SQL Server driven programming with Python GUI. Deliberately designed for various levels of programming skill, this book is suitable for students, engineers, and even researchers in various disciplines. There is no need for advanced programming experience, and school-level programming skills are needed. In the first chapter, you will learn to use several widgets in PyQt5: Display a welcome message; Use the Radio Button widget; Grouping radio buttons; Displays options in the form of a check box; and Display two groups of check boxes. In chapter two, you will learn to use the following topics: Using Signal / Slot Editor; Copy and place text from one Line Edit widget to another; Convert data types and make a simple calculator; Use the Spin Box widget; Use scrollbars and sliders; Using the Widget List; Select a number of list items from one Widget List and display them on another Widget List widget; Add items to the Widget List; Perform operations on the Widget List; Use the Combo Box widget; Displays data selected by the user from the Calendar Widget; Creating a hotel reservation application; and Display tabular data using Table Widgets. In chapter three, you will

learn: How to create the initial three tables project in the School database: Teacher, Class, and Subject tables; How to create database configuration files; How to create a Python GUI for inserting and editing tables; How to create a Python GUI to join and query the three tables. In chapter four, you will learn how to: Create a main form to connect all forms; Create a project will add three more tables to the school database: Student, Parent, and Tuition tables; Create a Python GUI for inserting and editing tables; Create a Python GUI to join and query over the three tables. In chapter five, you will join the six classes, Teacher, TClass, Subject, Student, Parent, and Tuition and make queries over those tables. In chapter six, you will create dan configure database. In this chapter, you will create Suspect table in crime database. This table has eleven columns: suspect_id (primary key), suspect_name, birth_date, case_date, report_date, suspect_status, arrest_date, mother_name, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for this table. In chapter seven, you will create a table with the name Feature_Extraction, which has eight columns: feature_id (primary key), suspect_id (foreign key), feature1, feature2, feature3, feature4, feature5, and feature6. The six fields (except keys) will have VARBINARY(MAX) data type. You will also create GUI to display, edit, insert, and delete for this table. In chapter eight, you will create two tables, Police and Investigator. The Police table has six columns: police_id (primary key), province, city, address, telephone, and photo. The Investigator table has eight columns: investigator_id (primary key), investigator_name, rank, birth_date, gender, address, telephone, and photo. You will also create GUI to display, edit, insert, and delete for both tables. In the last chapter, you will create two tables, Victim and Case_File. The Victim table has nine columns: victim_id (primary key), victim_name, crime_type, birth_date, crime_date, gender, address, telephone, and photo. The Case_File table has seven columns: case_file_id (primary key), suspect_id (foreign key), police_id (foreign key), investigator_id (foreign key), victim_id (foreign key), status, and description. You will create GUI to display, edit, insert, and delete for both tables as well.

The far side of the Moon, also called the "dark side of the Moon" was unknown to humanity until the Luna and Lunar Orbiter pictures were returned to Earth. This wonderful book contains beautiful photographs and newly-assembled mosaic images of the far side of the Moon, cleaned of transmission, imaging stripes and processing artifacts by today's computer technology. Byrne's superb analysis documents the appearance of the features of the far side with beautiful pictures from Lunar Orbiter. Until now, the far side Lunar Orbiter photos have only been available with strong reconstruction lines, but appear here for the first time as complete photographs, unmarred by imaging and processing artifacts.

[Copyright: 1c882c3c8b224d931dd269151ccc0836](http://www.copyright.com/1c882c3c8b224d931dd269151ccc0836)