Programming in Visual Basic

GUI Environment:

Introduction to graphical user interface (GUI),

The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

The graphical user interface, developed in the late 1970s by the Xerox Palo Alto research laboratory and deployed commercially in Apple's Macintosh and Microsoft's Windows operating systems, was designed as a response to the problem of inefficient usability in early, text-based command-line interfaces for the average user.

Graphical user interfaces would become the standard of user-centered design in software application programming, providing users the capability to intuitively operate computers and other electronic devices through the direct manipulation of graphical icons such as buttons, scroll bars, windows, tabs, menus, cursors, and the mouse pointing device. Many modern graphical user interfaces feature touch-screen and voice-command interaction capabilities.

Programming language

A programming language is a set of commands, instructions, and other syntax use to create a software program. Languages that programmers use to write code are called "high-level languages." This code can be compiled into a "low-level language," which is recognized directly by the computer hardware.

High-level languages are designed to be easy to read and understand. This allows programmers to write source code in a natural fashion, using logical words and symbols. For example, reserved words like function, while, if, and else are used in most major programming languages. Symbols like <, >, ==, and != are common operators. Many high-level languages are similar enough that programmers can easily understand source code written in multiple languages.

Examples of high-level languages include C++, Java, Perl, and PHP. Languages like C++ and Java are called "compiled languages" since the source code must first be compiled in order to

run. Languages like Perl and PHP are called "interpreted languages" since the source code can be run through an interpreter without being compiled.

Procedural Language

A procedural language is a type of computer programming language that specifies a series of well-structured steps and procedures within its programming context to compose a program. It contains a systematic order of statements, functions and commands to complete a computational task or program.

Procedural language is also known as imperative language. A procedural language, as the name implies, relies on predefined and well-organized procedures, functions or sub-routines in a program's architecture by specifying all the steps that the computer must take to reach a desired state or output.

The procedural language segregates a program within variables, functions, statements and conditional operators. Procedures or functions are implemented on the data and variables to perform a task. These procedures can be called/invoked anywhere between the program hierarchy and by other procedures as well. A program written in procedural language contains one or more procedures.

Object Oriented Language

Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

OOP focuses on the objects that developers want to manipulate rather than the logic required manipulating them. This approach to programming is well-suited for programs that are large, complex and actively updated or maintained. This includes programs for manufacturing and design, as well as mobile applications; for example, OOP can be used for manufacturing system simulation software.

Event Driven Program

An event-driven program is one that largely responds to user events or other similar input. The concept of event-driven programming is an important one in application development

and other kinds of programming, and has spawned the emergence of event handlers and other resources.

The idea in event-driven programming is that the program is designed to react. It reacts to specific kinds of input from users, whether it's a click on a command button, a choice from a drop-down list, an entry into a text box, or other kinds of user events.

Compiling a Program

Compile is the creation of an executable program from code written in a compiled programming language. Compiling allows the computer to run and understand the program without the need of the programming software used to create it. When a program is compiled it is often compiled for a specific platform (e.g., IBM platform) that works with IBM compatible computers, but not other platforms (e.g., Apple platform).

The first compiler was developed by Grace Hopper while working on the Harvard Mark I computer. Today, most high-level languages include a compiler or have toolkits available to compile the program. Two popular compilers are Eclipse for Java and gcc command for C and C++. Depending on how big the program is, it should take a few seconds or minutes to compile. If no errors are encountered while being compiled, an executable file is created.

Debugging a Program

Debugging is the process of detecting and removing of existing and potential errors (also called as ‘bugs’) in a software code that can cause it to behave unexpectedly or crash. To prevent incorrect operation of a software or system, debugging is used to find and resolve bugs or defects.

To debug a program, user has to start with a problem, isolate the source code of the problem, and then fix it. A user of a program must know how to fix the problem as knowledge about problem analysis is expected. When the bug is fixed, then the software is ready to use.

Running the programs

Execution is the process of running a program or the carrying out of the operation called for by an instruction.

Controls:

Introduction to controls textboxes

Textbox controls offer a natural way for users to enter a value in your program. For this reason, they tend to be the most frequently used controls in the majority of Windows applications. Text box controls allow entering text on a form at runtime. By default, it takes a single line of text, however, we can make it accept multiple texts and even add scroll bars to it.

Frames: -

We usually use the frame control to group controls together into a recognizable group. This control appears as a box with a label at upper left. We can make the controls in a frame into a functional group a swell, such as when we group option buttons together. When we add option buttons to a frame, those buttons function in concert, when we click one, all the others are deselected.

Check boxes

A list of choices can be made using check By ticking the CheckBox the value is set to True. This control can also be grayed when the state of the CheckBox is unavailable, When we place a CheckBox control on a form, we have to write its Caption property to a descriptive string. We might sometimes want to move the little check box to the right of its caption, which we do by setting the Alignment property to 1-Right Justify, but in most cases the default setting is OK. If we want to display the control in a checked state, we set its Value property to 1-Checked right in the Properties window, and we set a grayed state with 2-Grayed.

The only important event for CheckBox controls is the Click event, which fires when either the user or the code changes the state of the control. In many cases, we don't need to write code to handle this event. Instead, we just query the control's Value property when our code needs to process user choices. we usually write code in a CheckBox control's Click event when it affects the state of other controls. For example, if the user clears a check box, we might need to disable one or more controls on the form and reenable them when the user clicks on the check box again. This is how you usually do it

Option buttons,

A radio button or option button is a **graphical control element that allows the user to choose only one of a predefined set of mutually exclusive options**. Option buttons, also called radio buttons, are typically used in a group of two or more. At any one time, only one button in the group can be "on". Clicking an option button turns it "on" and turns all other buttons in the group "off".

Option button groups operate in a container control, such as a frame. Therefore, different sets of option button groups should be placed in their own frame on the form. If a group of option buttons is not contained within a frame, then the form itself acts as their container.

In code, to perform an action based on which option button the user clicked, do one of two things:

(1) To perform an action as soon as the user clicks an option button, place code in the Click event of the option button.

(2) To perform a "delayed" action, such as when the user clicks a command button, check the **Value** property of the option buttons in the group. If the Value = True, then the button is "on".

Images

Image box is **only used to display images** (such as Bitmap),

Setting

Borders and styles,

The shape control:

The Shape control is a graphical control. We can use this control to draw predefined coloured and filled shapes, including rectangles, squares, ovals circles, rounded rectangles or rounded squares.

We use the shape control at design time to draw shapes in a form. The shape control is a little like the frame control, however, shapes can not act as control containers.

The line control

The line control is a graphical control. We use it to display horizontal, vertical or diagonal likes in a form. We can use these controls at design time as a design element or at runtime to alter the original line we drew.

working with multiple controls and their properties,

designing the user interface,
keyboard access,
tab controls,
default & cancel property,
coding for controls.

(4L)
Operations:

Data types

Data Types	Storage capacity (in byte)
Boolean	2
Byte	1
Currency	8
Date	8
Decimal	12
Double	8
Integer	2
Long	4
Object	4
Single	4

String	N/A
User-defined data types	N/A
Variant	N/A

Constants:

Constants do not change their values. The syntax to declare a constant is

[Public | Private] Const constname [As type] = expression

The Public keyword is used at the module level to make a constant global. The Private keyword is used at the module or form level to declare constants that are private. The constname identifier is the actual name of the constant. The type identifier is the data type of the constant. The expression identifier holds the value we want for this constant. It may be a literal, other constant, or any combination that includes all arithmetic or logical operation.

named & intrinsic

declaring variables

scope of variables,

val function,
arithmetic operations,
formatting data.

Decision Making

If statement

If-Else statement is a decision making statement. The syntax is

```
If condition Then
        Statement
Else
        Statement
End if
```

```
If condition Then
        Statement
ElseIf Condition
        Statement
Elseif Condition
        Statement
Else
```

comparing strings,

compound conditions (and, or, not),

Nested if statements

```
If condition Then
        If Condition Then
                Statement
        Else
                Statement
        End if
End if
```

Case structure

When our program can handle multiple values of a particular variable then instead of using IF….Else statements to handle them, we should consider Select Case structure. This structure tests an expression,seeing which of several cases it matches and execute the corresponding code. The syntax is :

```
        Select Case testexpression
                Case expressionlist-n
                        Statements
                Case Else
                        Statement
        End Select
```

using if statements with option buttons & check boxes,

displaying message in message box,

testing whether input is valid or not**.**

(5L)

Forms Handling :

Multiple forms creating,

adding, removing forms in project,

hide, show method,

load,unload statement,

me keyword,

referring to objects on a different forms.

Iteration Handling: Loop means the repetition of statements for a finite number of times. There are three types of loop used in visual basic

Do/loop:

```
Do [{while | Until} condition]
        Statements
loop
```

```
Do
        Statements
Loop [{While | Until} condition]
```

for/next loops

```
For Index = start to end [Step step ]
        Statements
```

The While Loop

```
While condition
        Statements
Wend
```

Using msgbox function

We can use MsgBox() to display a message to the user and get a return value corresponding to one of the buttons in the message box. The syntax is

```
MsgBox (prompt [. Buttons] [. Title] [.helpfile, context])
```

The ***prompt*** argument holds the string displayed as the message in the dialog box. The maximum length of prompt is approximately 1,024 characters. The ***buttons*** argument specifies what to put into the message box. The ***title*** parameter holds the string displayed in the title bar of the dialog box. The ***Helpfile*** argument is a string that identifies the Help file to use to provide context-sensitive Help for the dialog box.

using string function.