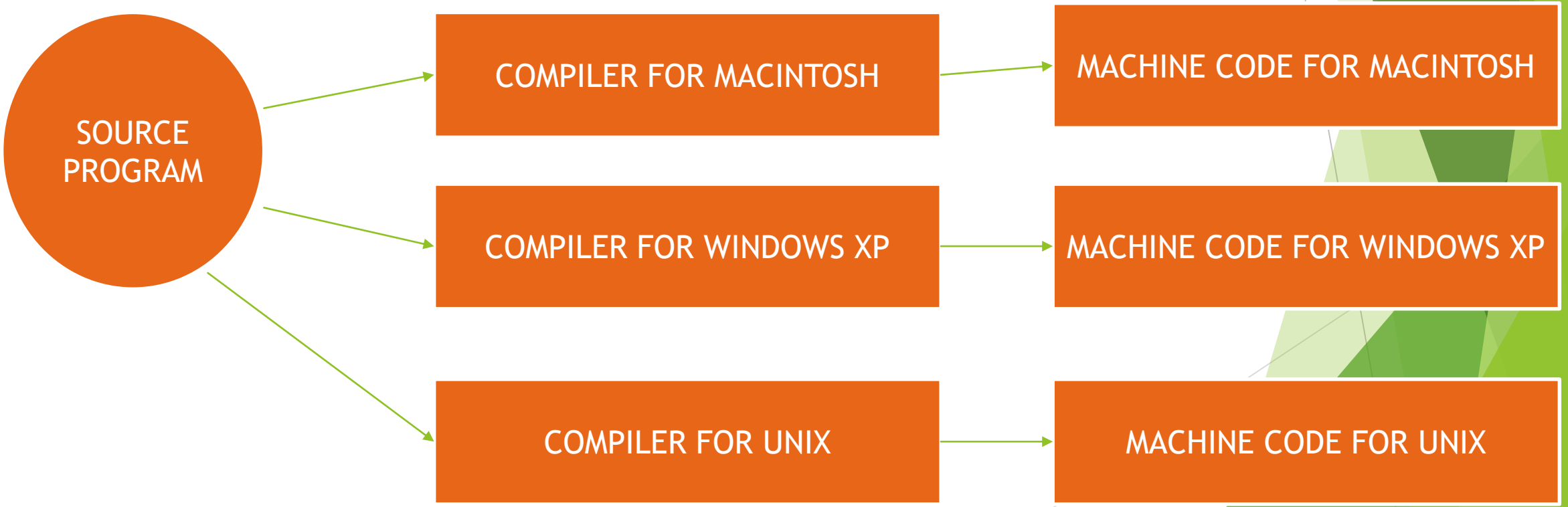
INTRODUCTION TO JAVA

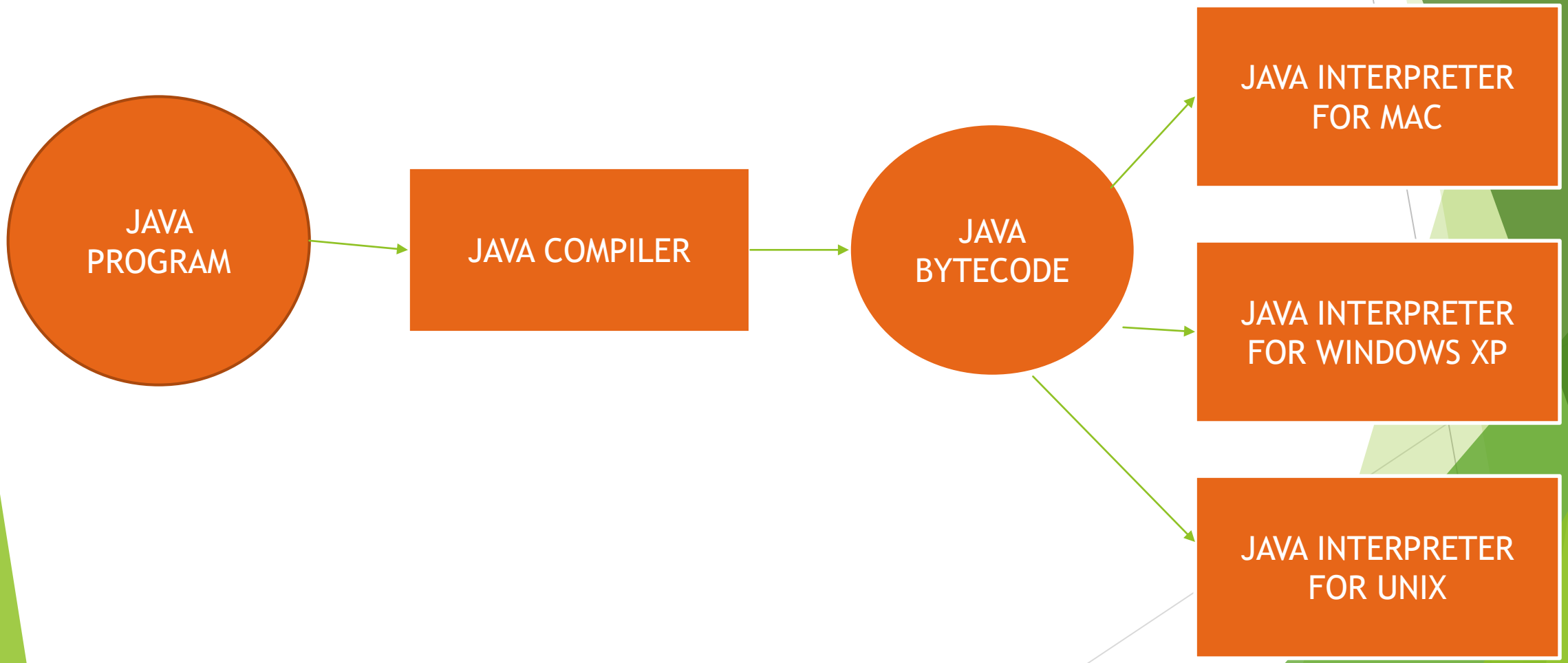
Java is a platform independent programming language created by James Gosling from Sun Microsystems in 1991

JAVA COMPILATION

ORDINARY COMPILATION:

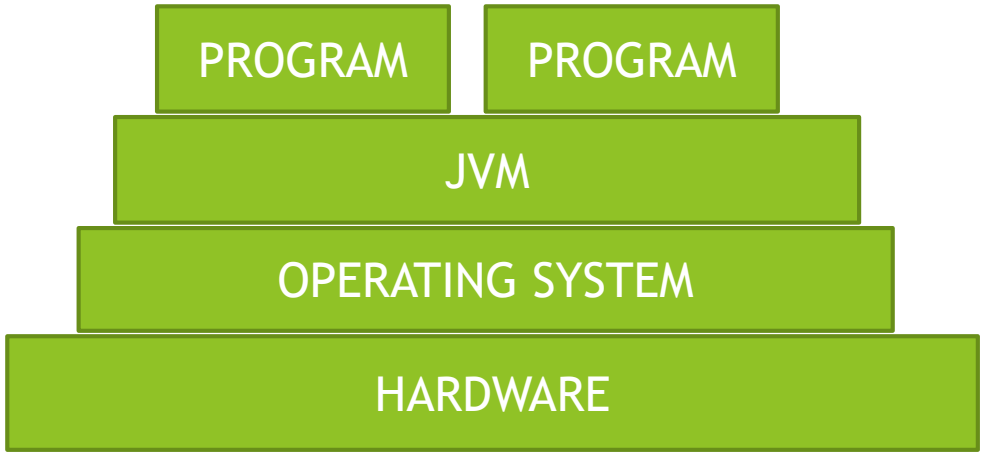


JAVA COMPILATION



JAVA VIRTUAL MACHINE (JVM)

Java programs are compiled by the JAVA COMPILER into BYTECODE. The Java virtual machine interprets this bytecode and executes the Java Program.



CHARACTERISTICS OF JAVA

1) Write Once Run Anywhere (WORA).

2)Light weight code.

3)Security.

4)Built-in-graphics.

5)Object oriented language.

6)Supports Multimedia.

7)Platform Independent.

8)Open product.

RAPID APPLICATION DEVELOPMENT(RAD)

Rapid development of application is possible through RAD tools. The RAD tools are the tools that enable one to create applications in shorter time compared to other conventional language.

Definition: RAD describes a method of developing software through the use of pre-programmed tools.

INTRODUCTION TO NETBEANS JAVA IDE

NETBEANS IDE is a free ,open source ,cross platform IDE(Integrated Development Environment) with built-in support for JAVA programming.

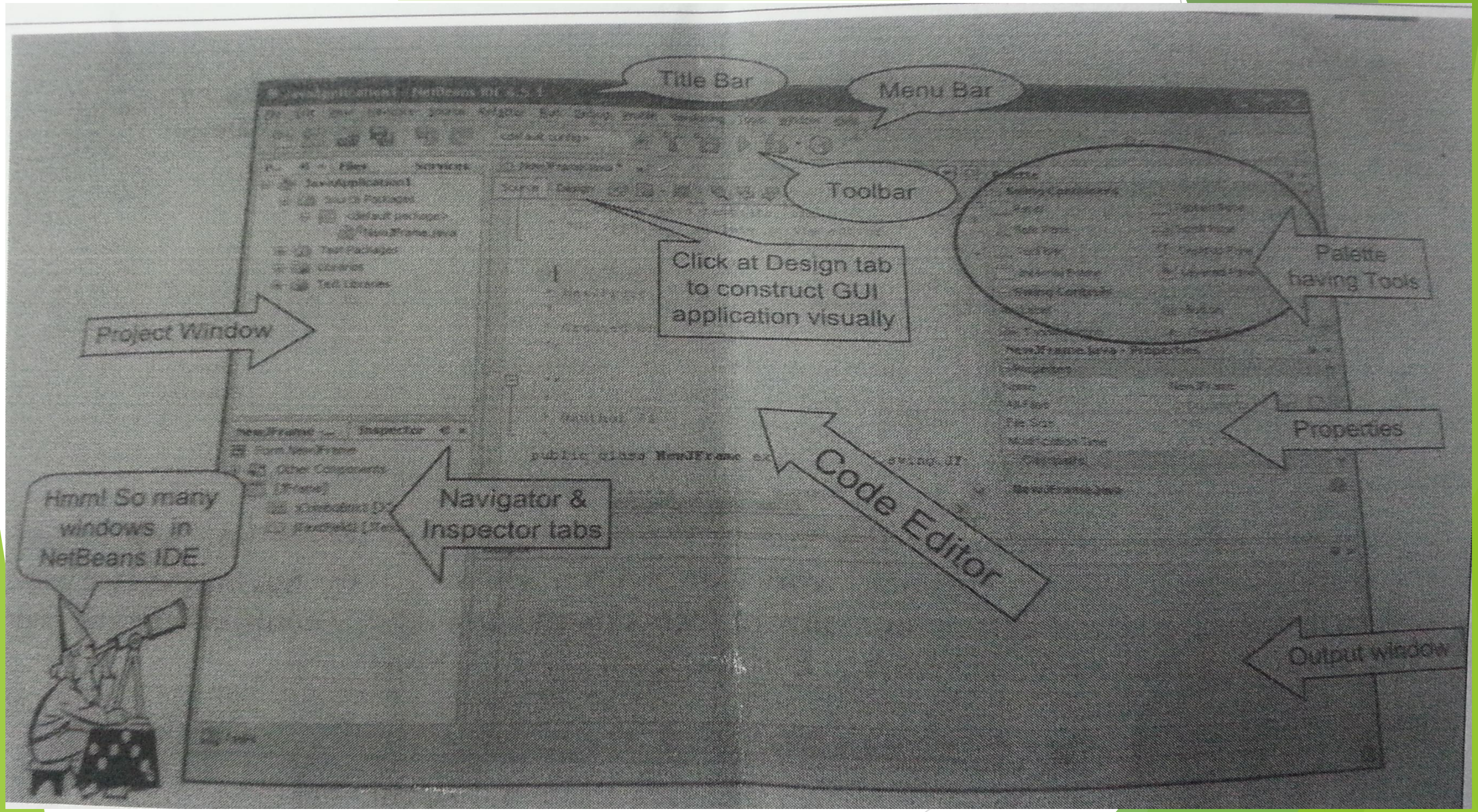
NetBeans offers:

1. Drag and drop GUI creation.
2. Excellent editing(advanced source code editor)
3. Web services
4. Excellent debugging
5. Wizard, code generation and management tools

STEPS TO PROGRAM JAVA IN NETBEANS

- Open NETBEANS IDE.
- Create a new project (File -> New project)
- Click on Java and Java application options
- Enter the project (name should be related to the program)
- Click on Finish button.
- Project name will be displayed

VISUAL TOUR OF NETBEANS

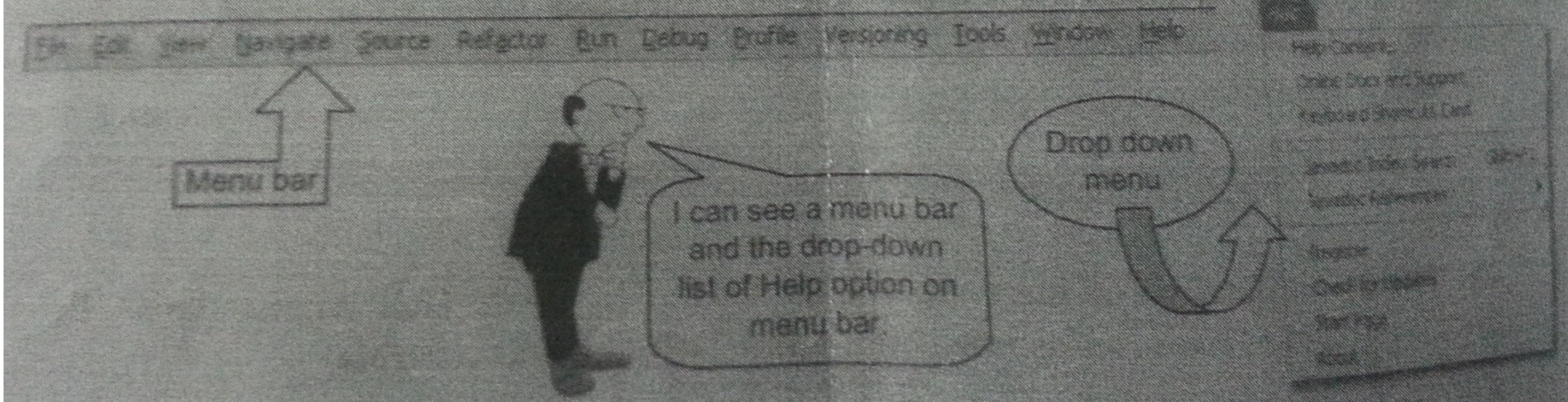


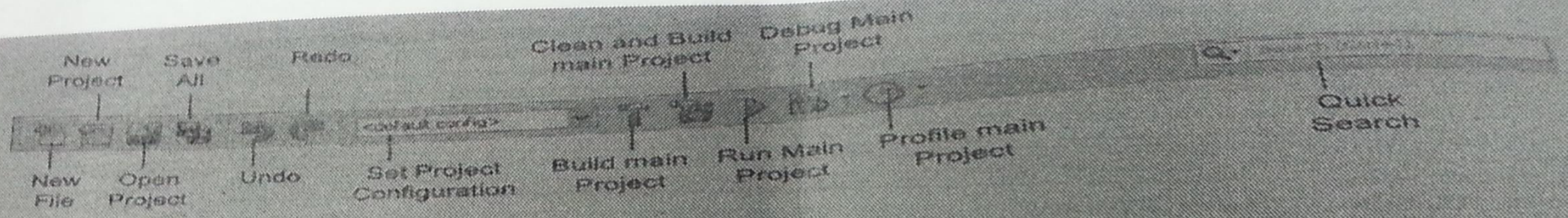
Title Bar

The title bar displays the title of the application. By default, NetBeans will give names as JavaApplication1, JavaApplication2, ... etc., to your project. Notice in following figure, it has given title *JavaApplication1* to your application.

Menu Bar and Pull-down Menus

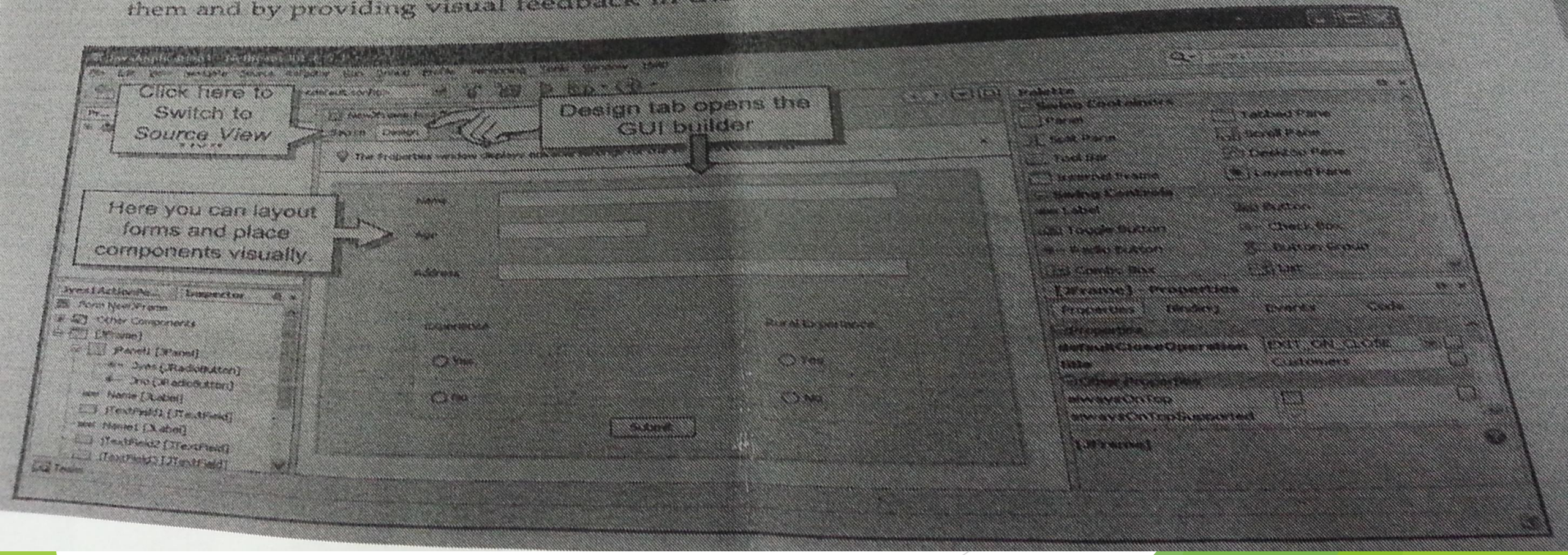
You are familiar with menu bars as you have worked with MS-Windows operating system. A menu bar is displayed directly below that title bar and includes a lot of options. Each option on the menu bar has a *drop-down list* of items (known as **Pull-down menus**) that help you perform various tasks.

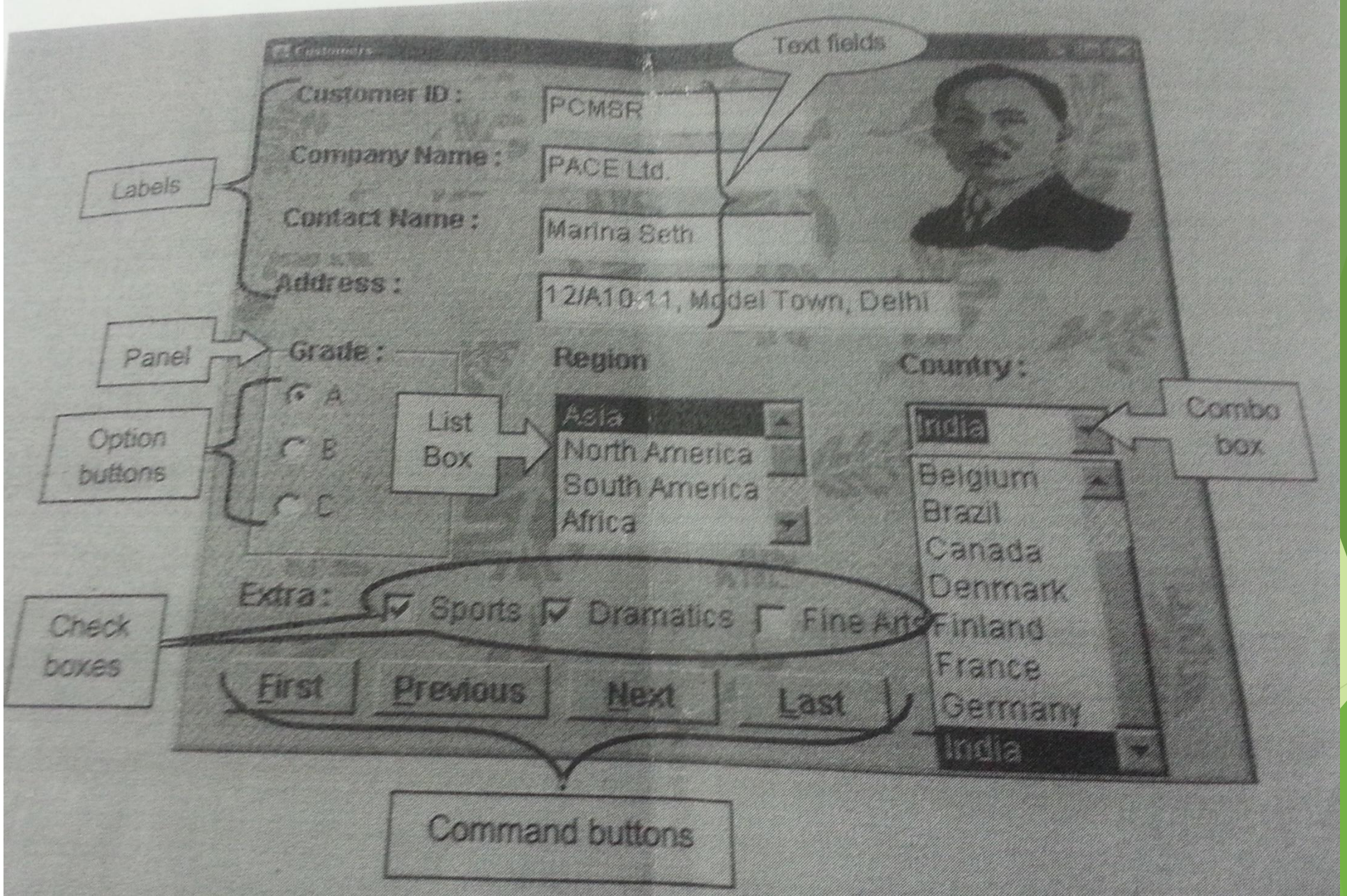




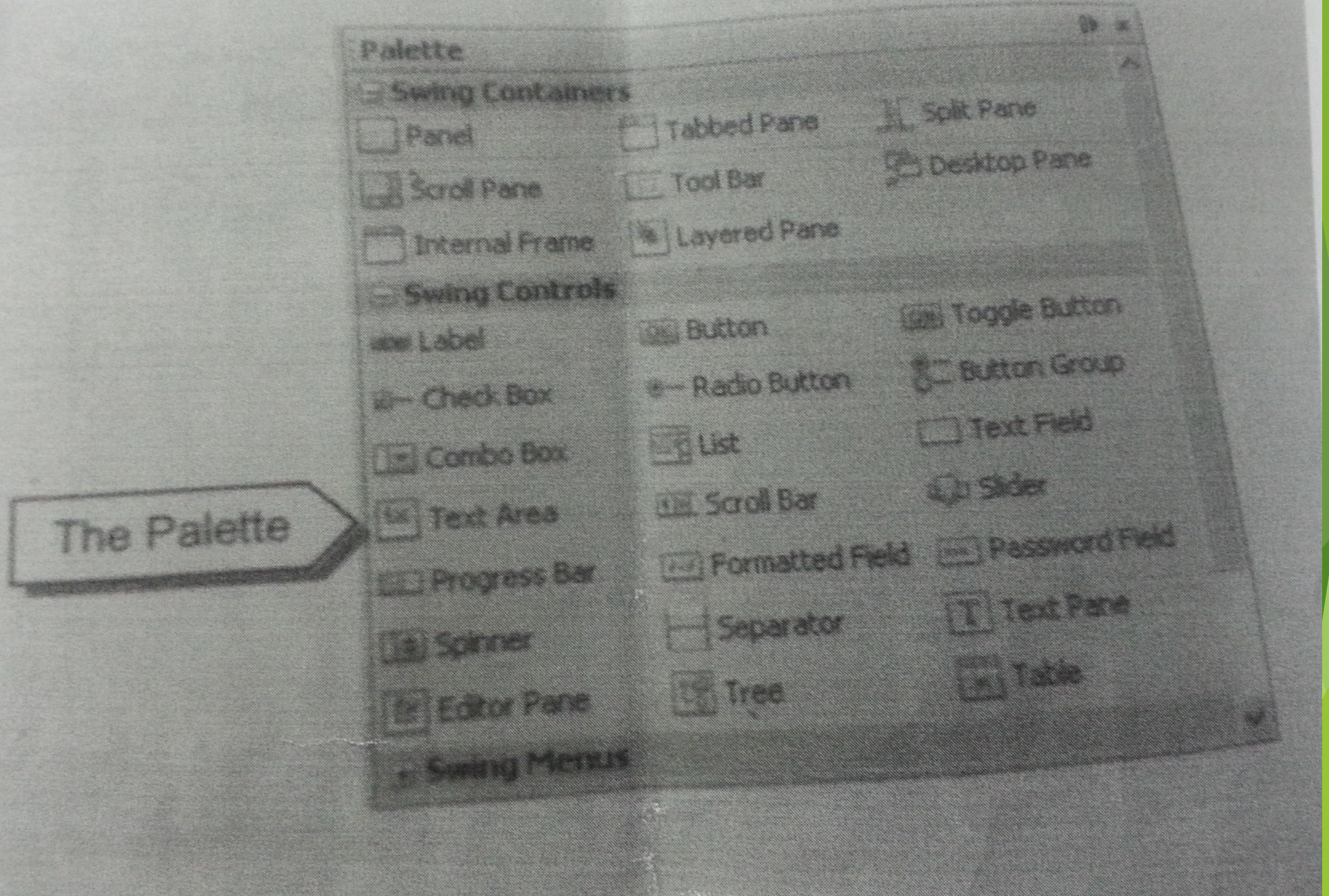
GUI builder

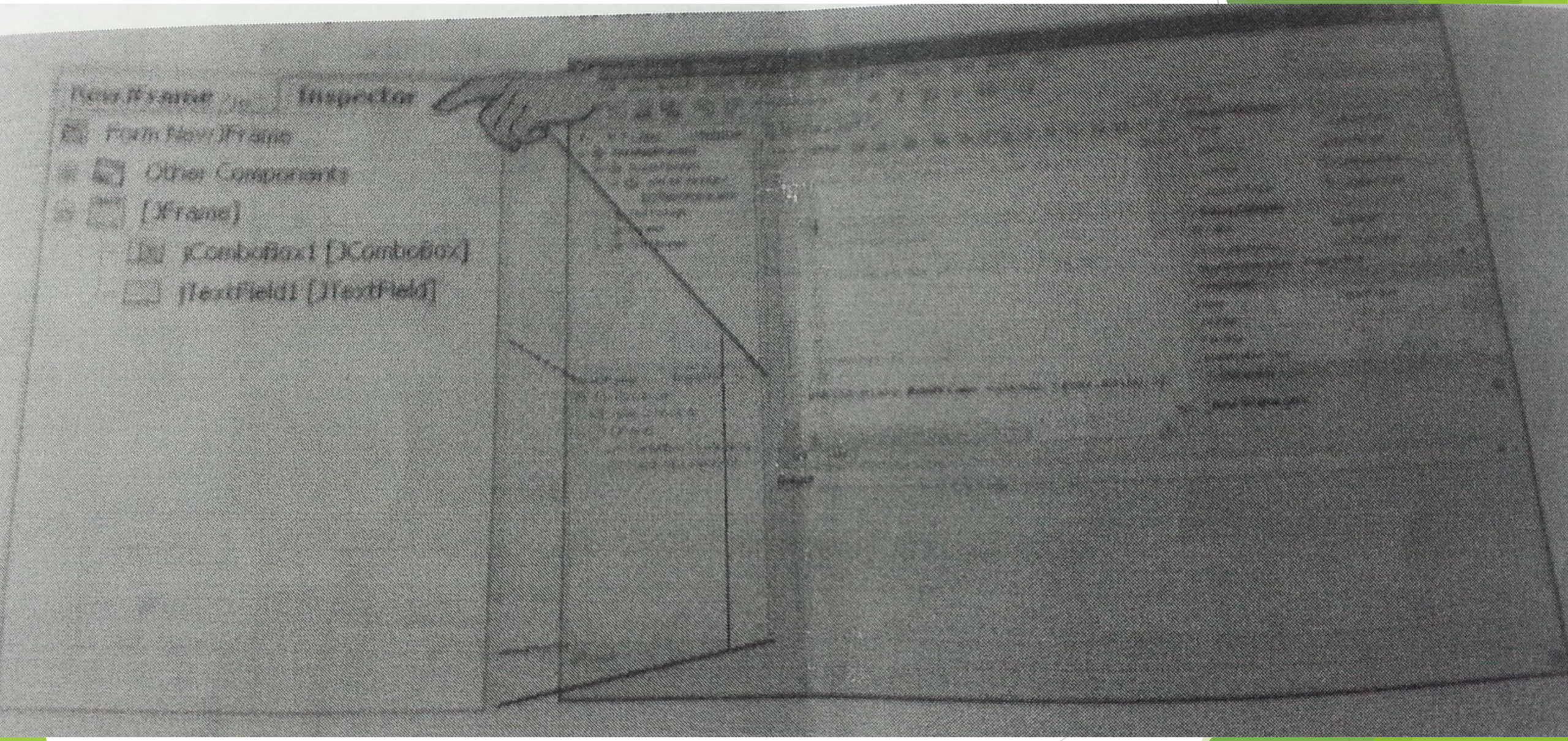
Also called **Design Area** or **Design Space**. The Design Area is where you will visually construct your GUI. It is the primary workspace within which GUI design takes place in the IDE. The GUI Builder enables you to layout forms by placing components where you want them and by providing visual feedback in the form of guidelines.





Different types of graphical controls.


Palette
Swing Containers
Panel
Tabbed Pane
Split Pane
Scroll Pane
Tool Bar
Desktop Pane
Internal Frame
Layered Pane
Swing Controls
Label
Button
Toggle Button
Check Box
Radio Button
Button Group
Combo Box
List
Text Field
Text Area
Scroll Bar
Slider
Progress Bar
Formatted Field
Password Field
Spinner
Separator
Text Pane
Editor Pane
Tree
Table
Swing Menus
The Palette


NewJFrame
Inspector
From NewJFrame
Other Components
[JFrame]
jComboBox1 [JComboBox]
jTextField1 [JTextField]

3.8 SETTING PROPERTIES

The controls/objects that you draw on your frame/ window have some properties associated with them. Different controls have different properties associated with them.

The *Properties window* (see Fig. 3.9) provides an easy way to set properties for all objects on a frame/window. To open the **Properties window** (if it is not open), choose the **Properties** command from the **Window** menu. You may also press the shortcut key for it which is : *Control + Shift + 7*.

The Properties window consists of the following elements :

- **Title box.** Displays the name of the object for which you can set properties.
- **Properties list.** The left column under *Properties* tab, displays all of the properties for the selected object. You can edit and view settings in the right column.

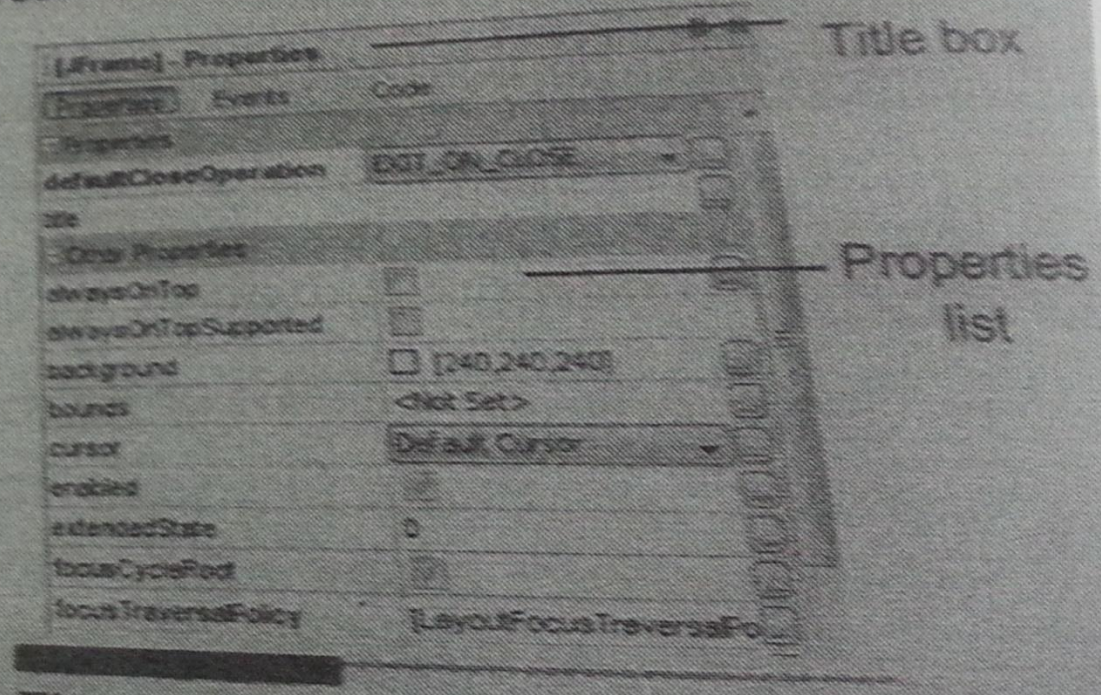


Figure 3.9

To set properties from the *Properties window* :

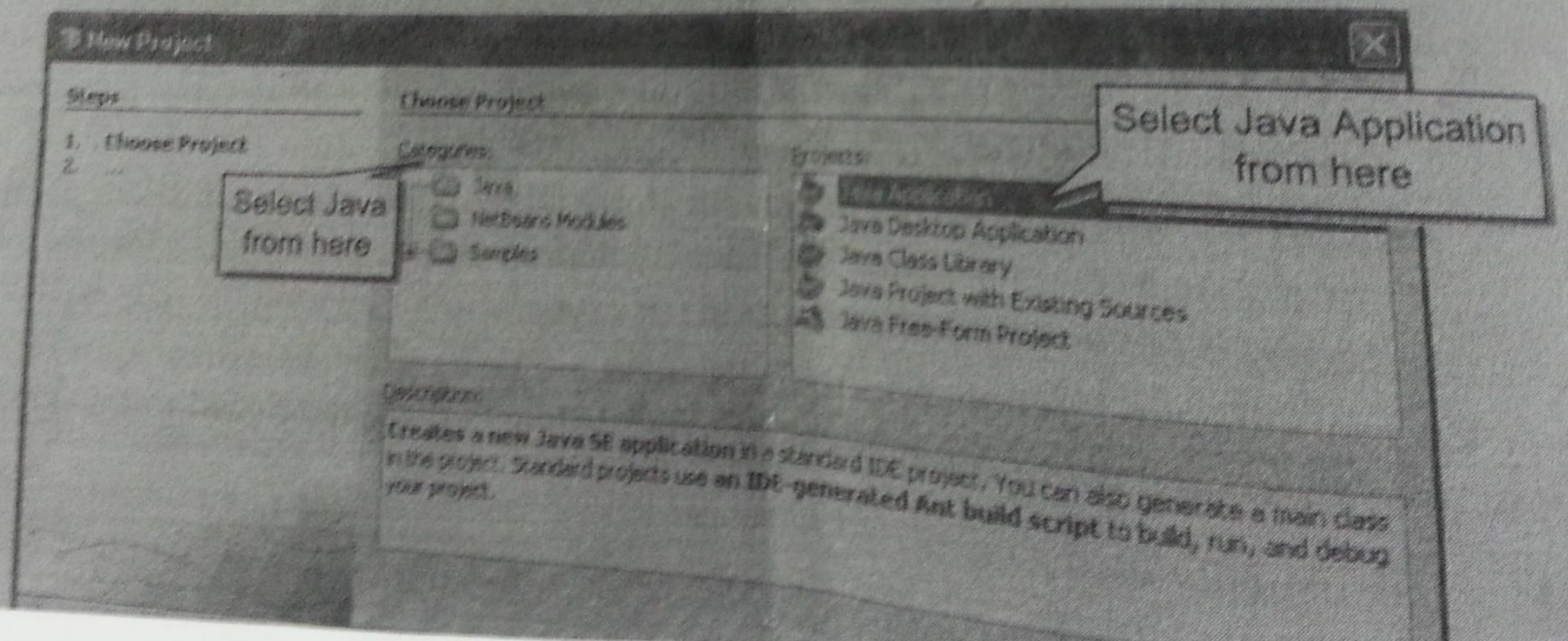
1. In the *Properties window*, from the *Properties list*, select the name of a property.
2. Type in the right column, or click ( ) to type or select the new property setting

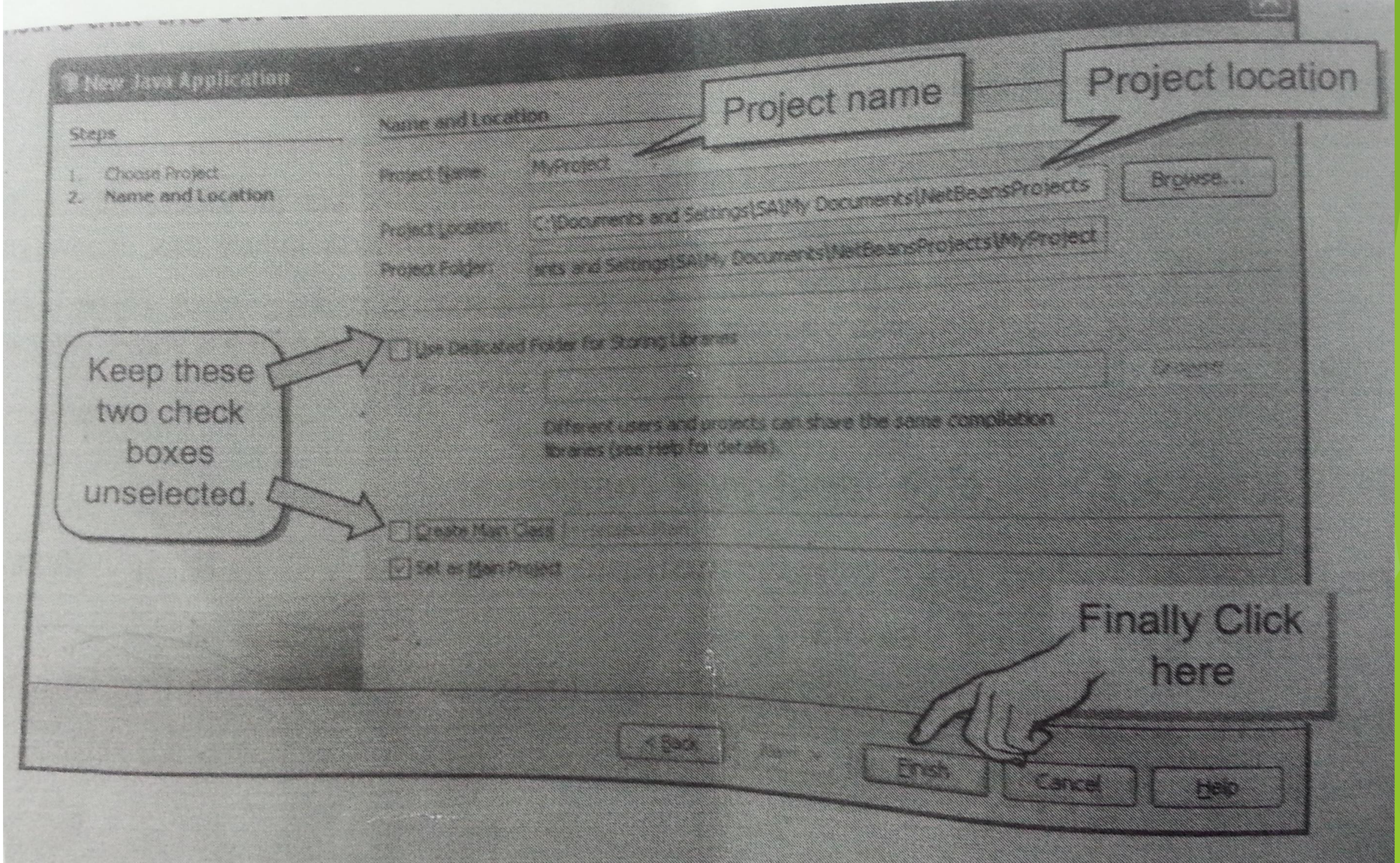
3.10 CREATING A PROJECT IN NETBEANS GUI BUILDER

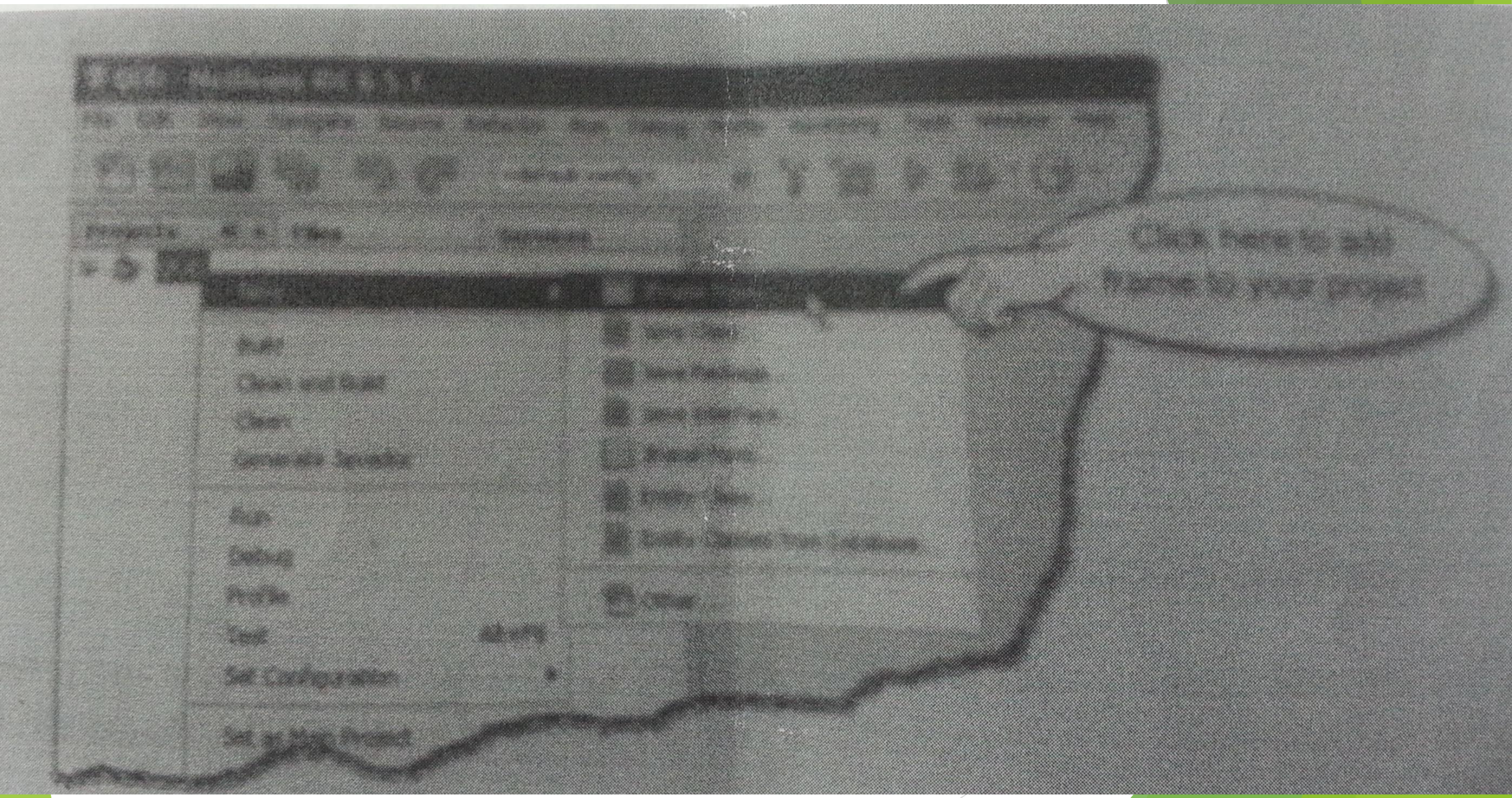
All Java development in the IDE takes place within projects, we first need to create a new project within which to store sources and other project files. An IDE project is a group of Java source files plus its associated *meta data* (data about data), including project-specific properties files, and many other related files. A Java application may consist of one or more IDE projects.

To create a new GUI application project :

1. Click **File → New Project** command. Alternately, you can click the *New Project* icon ( ) in the IDE toolbar or press Ctrl + Shift + N.
2. In the **Categories** pane, select the **Java** node and in the **Projects** pane, choose **Java Application**. Click **Next.** (see below)




New Java Application
Steps
1. Choose Project
2. Name and Location
Name and Location
Project name
Project location
Project Name:
MyProject
Project Location:
C:\Documents and Settings\SA\My Documents\NetBeansProjects
Browse...
Project Folder:
Keep these two check boxes unselected.
Create Main Class
Set as Main Project
Finally Click here
< Back
Finish
Cancel
Help


Build
Clean and Build
Clean
Generate Javadoc
Run
Debug
Profile
Test
Set Configuration
Set as Main Project
Click here to set
Name to your project

CHAPTER-2: JAVA CHARACTER SET

CHARACTER SET is a set of valid characters that a language can recognize.

A character represents any letter, digit or any other sign.

JAVA uses UNICODE character set.

UNICODE is a two-byte character code that has characters representing almost all characters in all language.

UNICODE is similar to ASCII character set.

UNICODE character is represented by using escape sequence(\u) followed by a four digit hexadecimal number.

for example:

\u00AE © The copyright symbol

\u0022 " The double quote

\u0394 Δ The capital Greek letter delta

TOKENS

Tokens are the smallest individual unit in a program.

Types of Tokens:

1) Keywords
2) Identifiers
3) Literals
4) Punctuators
5) Operators.

KEYWORDS:

Keywords are the words that convey special meaning to the language compiler.

Eg:

void,if,return,while,public,float,switch,else, byte,class,char,goto.. etc.

IDENTIFIERS:

Identifiers are the fundamental building blocks of a program.

RULES FOR FORMING IDENTIFIERS:

1) Identifiers can have alphabets,digits,underscore and doller sign characters.

2) Identifiers must not be keywords or Boolean literal.

3)Identifiers must not begin with digit.

4) Identifiers can be of any length.

VALID IDENTIFIERS:

Myfile
_as
a_z
file1
$1_to_$10
date23_5_16

INVALID IDENTIFIERS:

DATA-REC
26ISWK
VOID
MY.FILE

LITERALS:

Literals(often referred to as constants) are data items that never change their value during a program run.

Java allows:

1) Integer literals:

IL are the whole numbers without any fractional part.

There are three types of integer literals:

1) Decimal (base 10)
2) Octal (base 8)
3) Hexadecimal(base 16)

2) Character literals:
A character literal is one character enclosed in single quotes.
Eg: ‘z’

3) Floating literals:
Floating literals are called as real literals. Real literals are numbers having fractional part.
The two forms of real literals are fractional form and exponent form.

VALID REAL LITERALS:
2.0,-12.89,-0.0234

INVALID REAL LITERALS:
89.
+89/3
34,780.76
45,890

STRING LITERALS:
Multiple character constants are called as string literals.
Eg: “ISWK”

PUNCTUATORS:
The following characters are used as punctuators(also known as seperators)
[ ] ( ) { } , ; : * = #

OPERATORS:

Operators are tokens that trigger some computation when applied to variables in an expression.

1) Unary operators:

Unary operators are the operators that require one operator to act upon.

&-address operator

+ unary plus

- unary minus

++ increment operator

-- decrement operator

BINARY OPERATORS:
Binary operators are the operators that act upon two operands to operate upon.

Arithmetic operators:
+ Addition
- subtraction
* multiplication
/ division
% reminder/modulus

Logical operator:

&& logical AND

|| logical OR

Relational operator:

< less than

> greater than

<= less than or equal to

>= greater than or equal to

== equal to

!= not equal to

DATA TYPES:

Data types are the means to identify the type of data and associated operations of handling it.

There are two types of data types

1) Primitive data type:

Java provides 8 primitive data types

Byte,short,int,long – Numeric Integral primitive types.

float,double – Fractional primitive types

char – Character primitive types

Boolean – Boolean primitive types

2) Reference data type:

Reference data type are constructed from primitive data types.

These are classes,arrays and interface.

Variables:
Variables represent named storage location whose values can be manipulated during program run.

Declaration of variable:
Syntax:
type variablename;
eg: int a;

Initialization of variables:
eg: int a=10;

Text Interaction Methods:

There are 3 types of text interaction methods in Java.

1) getText() method:

A getText() method returns the text currently stored in text based GUI component.

Swing components that support getText() method are Text field,Text area,Button,Label,Check box and Radio button.

Eg: jTextField1.getText()

2) setText() method:

A setText() method stores or changes text in a text based GUI component.

Swing components that support setText() method are Text field,Text area,Button,Label,Check box and Radio button.

Eg: jTextField1.setText(“class X”)

3) Parse…() method:

Parse…() method helps to parse string into different numeric types.

Byte.parseByte(String s)-converts a string into a byte type value.
Short.parseShort(String s) -converts a string into a short type value.
Integer.parseInt(String s) -converts a string into a integer type value.
Long.parseLong(String s) -converts a string into a long type value.
Float.parseFloat(String s) -converts a string into a float type value.
Double.parseDouble(String s) -converts a string into a double type value.

Eg: String a=jTextField1.getText();
int b=Integer.parseInt(a);

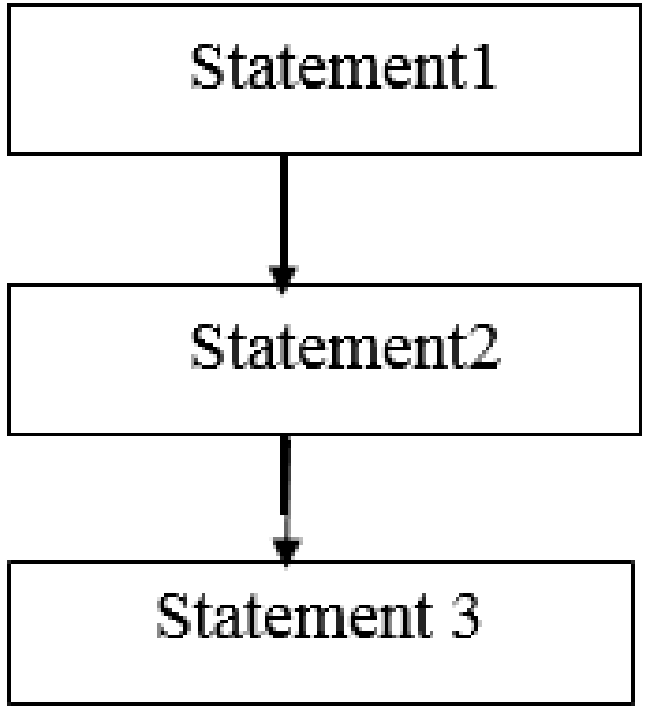
PROGRAMMING CONSTRUCTS

The statements inside your source files are generally executed from top to bottom, in the order that they appear. Control flow statements, however, breakup the flow of execution by employing decision making, looping, and branching, enabling your program to conditionally execute particular blocks of code.

DIFFERENT TYPES OF PROGRAMMING CONSTRUCT:

1. **SEQUENCE**
2. **SELECTION**

SEQUENCE CONSTRUCT: Sequence construct means the statements are being executed sequentially. It is a default flow of statement from top to bottom.



SELECTION : A selection statement selects among a set of statements depending on the value of a controlling expression. They are also called as **Decision Making Statements**. They are:

if statements

if else statements

if statements: The if statement allows selection (decision making) depending upon the outcome of a condition. If the condition evaluates to true then the statement immediately following if will be executed and otherwise the first set of code after the end of the if statement (after the closing curly brace)will be executed.

Simple if:

The syntax of if statement is as shown below:

```
if (conditional expression)
{
Statement Block;
}
```

Example

```
int x = 10;
if( x < 20 ){
System.out.print("This is if statement");
}
```

This would produce the following result:

```
This is if statement
```

```
int x=Integer.parseInt(jTextField1.getText());

if(x>20)

jLabel3.setText("X is greater than 20");

else

jLabel3.setText("X is less than 20");
```

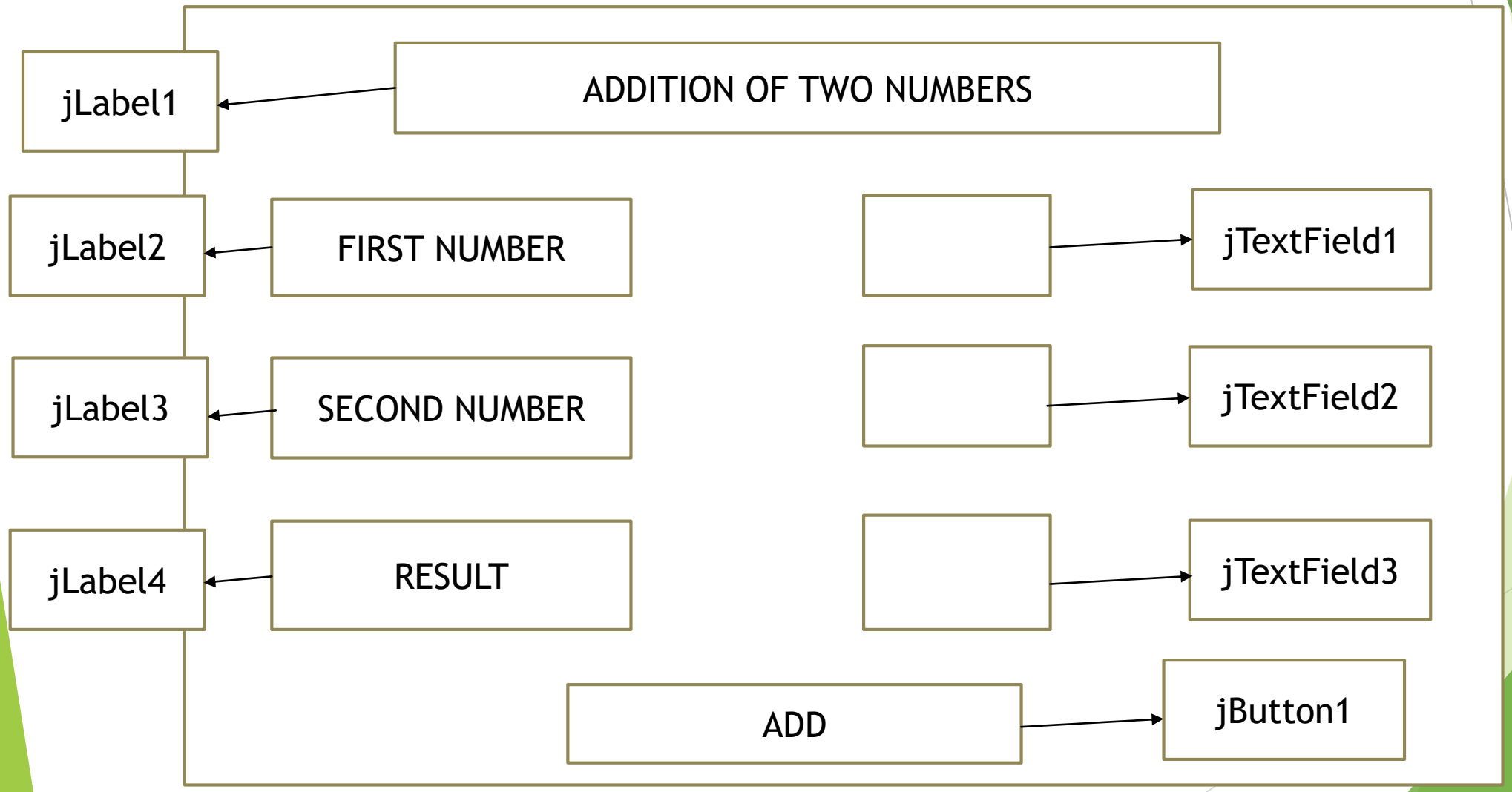
The if...else Statement: An if statement can be followed by an optional else statement, which executes when the Boolean expression is false. syntax of if-else statement is as shown below:

Syntax

```
if (conditional expression)
{
Statement Block;
}
else
{
Statement Block;
}
```

Example: int
x = 30; if(x
< 20){
System.out.print("This is if statement");
}
else
{
System.out.print("This is else statement");
}
This would produce the following result:
This is else statement

DESIGN A JFRAME TO ADD TWO NUMBERS



ON CLICK OF ADD BUTTON:

```
int num1=Integer.parseInt(jTextField1.getText());
int num2=Integer.parseInt(jTextField2.getText());
int num3=num1+num2;
jTextField3.setText(“”+num3);
```