UNIT – II

INTRODUCTION TO PROGRAMMING

Getting started with IDE

- **RAD**: Rapid Application Development is software programming technique that allows quick development of software application.
- Integrated Development Environment (IDE): It is a software tool to help programmer to edit, compile, interpret and debug the program in the same environment. i.e Eclipse,NetBeans, VB etc.
- Byte code: A byte code is machine instruction that the Java compiler generates and Java interpreter executes. When the compiler compiles a .java file, it produces a series of byte codes and stores them in a .class file. The Java interpreter (JVM) can execute the byte codes stored in the .class file.
- **JVM**: Java Virtual Machine (JVM) is a program which behaves as interpreter and translates the byte code into machine language as they go called just in time compilation.
- **Source Code**: The core program or text which is written in a language like C,C++ or Java is called source code.
- **Object Code:** The program which only is understood by the computer in the form of machine instructions or binary instructions called object code. In Java JVM is used to generate object code in the form of byte code.
- **GUI**: A graphical user interface (GUI) presents a pictorial interface to a program. GUI allows the user to spend less time trying to remember which keystroke sequences do what and spend more time using the program in a productive manner.

Programming Fundamentals

Token

The smallest individual unit in a program is known as Token. Java has the following types of tokens: *keyword, Identifier, literal, punctuators and operators*.

Keywords

Keywords are words that have a specific predefined meaning in Java. They cannot be used as variable names. They are also known as reserve words. Eg. void, private, if, while etc.

Literals:

Items having fixed data values are referred to as Literals. They are also known as Constants. Various types of literals available in Java are :

- Integer literals
- Floating literals
- Boolean literals
- Character literals
- String literals
- Null literals

Variable :

Variable is a named storage location in computer memory whose contents can change during a program run.

The characteristics of a variable are:

- (i) It has a name.
- *(ii)* It is capable of storing values.
- (iii) It provides temporary storage.
- *(iv)* It is capable of changing its value during program execution.

Punctuators:

The following nine ASCII charaters are the separators:

() { } [] ; , .

Operators: Operators are special symbols that perform specific operations on one, two, or three operands, and then return a result.

Operators	Precedence
postfix	expr++ expr
unary	++exprexpr+exprexpr~ !
multiplicative	* / %
additive	+ -
shift	<<>>>>
relational	<><= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive	OR ^
bitwise inclusive	OR
logical	AND &&
logical	OR
ternary	?:
assignment	= += =*= /= %= &= ^= = <<= >>>=

Data type states the way the values of that type are stored, and the range for that type.

Material Downloaded From SUPERCOP



Primitive Data Types:

The Java programming language is statically typed, which means that all variables must first be declared before they can be used.

A primitive type is predefined by the language and is named by a reserved keyword. The eight primitive data types supported by the Java programming language are:

- byte: The byte data type is an 8bit signed two's complement integer. It has a minimum value of 128and a maximum value of 127 (inclusive).
- short: The short data type is a 16bit signed two's complement integer. It has a minimum value of 32,768and a maximum value of 32,767 (inclusive).
- int: The int data type is a 32bitsigned two's complement integer. It has a minimum value of2,147,483,648and a maximum value of 2,147,483,647 (inclusive).
- long: The long data type is a 64bitsigned two's complement integer. It has a minimum value of 9,223,372,036,854,775,808and a maximum value of 9,223,372,036,854,775,807(inclusive).
- **float**: The float data type is a singleprecision32bitIEEE 754 floating point.
- double: The double data type is a doubleprecision64bitIEEE 754 floating point.
- boolean: The boolean data type has only two possible values: true and false. Use this data typefor simple flags that track true/false conditions.
- char: The char data type is a single 16bitUnicode character. It has a minimum value of '\u0000' (or 0) and a maximum value of '\uffff ' (or 65,535 inclusive).

Reference Data Types : These are constructed by using primitive data types, as per user need. Reference data types, as per user need. Reference data types store the memory address of an object. Class, store the memory address of an object.

Class, Interface and Array are the example of Interface Reference Data types.

parse methods: parse() methods helps to parse string into different numeric types. These are :

Method	Syntax	Usage
parseByte()	Byte.parseByte(string)	To convert a string value to byte type
parseShort()	Short.parseShort(string)	To convert a string value to type short
parseInt()	Integer.parseInt(string)	To convert a string value to Integer type
parseLong()	Long.parseLong()	To convert a string value to Long type
parseFloat()	Float.parseFloat()	To convert a string value to Float type
pareseDouble()	Double.parseDouble()	To convert a string value to Double type

Type Conversion:

The process of converting one predefined type into another is called Type Conversion.

These are of two types:

a) Implicit type conversion

b) Explicit type conversion

- Implicit Type Conversion: In this conversion java compiler converts all operands up to the type of largest datatype.
- > Explicit Type Conversion:

An explicit type conversion is user defined that forces an expression to be of specific type.

Flow of Control

• **Control Flow Statements**: 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.



1. Selection: A selection statement selects among a set of statements depending on the value of a controlling expression.

(a) 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 if the condition evaluates to false then the statements following the else clause will be executed.

```
(i) Simple if:
```

The syntax of if statement is as shown below: Syntax: if (conditional expression) { Statement Block; } (ii) if-else The syntax of if-else statement is as shown below: Syntax: if (conditional expression) { Statement Block;

> } else

s eise

Statement Block;

}

(iii) Nested if else

These control structures are used to test for multiple conditions as against the simple if statement which can be used to test a single condition. The syntax of nested if else is as follows:

Syntax:

```
if (conditional expression1)
{
  statements1;
  }
  else if (conditional expression2)
  {
  statements2;
  }
  else if (conditional expression3)
  {
  statements3;
  }
  else
  {
  statements4;
  }
```

(b) **switch**: This selection statement allows us to test the value of an expression with a series of character or integer values. On finding a matching value the control jumps to the statement pertaining to that value and the statement is executed, till the break statement is encountered or the end of switch is reached.

The syntax of the switch statement is as follows:

```
switch (Variable/Expression)
{
     case Value1 : statements1 ;
        break ;
     case Value2 : statements2 ;
        break ;
     default: statements3 ;
}
```

2. Looping: These statements are used to perform a set of instructions repeatedly while the condition is true.

(i) The syntax of the for loop is:

Syntax for(initialization; test expression; increment/decrement expression) { statements; }

(ii) **While loop**: The while loop is an entry-controlled loop. It means that the loop condition is tested before executing the loop body. If the loop condition is initially false, for the first iteration, then loop may not execute even once.

The syntax of the while loop is as follows:

```
Syntax
while(test expression)
{
loop body
}
```

(iii) **do while** : Do..While loop is an exit-controlled loop. In the do..while loop, the test occurs at the end of the loop. This ensures that the do..while loop executes the statements included in the loop body at least once.

The syntax of the loop is as follows: Syntax : do { loop body }while (test expression);

3. Jump:

 (i) break : The break is used to break from an enclosing do, while ,for or switch statement. Syntax: break;

(ii) **continue:** The continue statement stops the execution of the current iteration and causes control to begin with next iteration.

Syntax: continue;

(iii) return : Return is used to return value from the method Syntax: Return <value>;

Java IDE Programming – I , II & III

• Commonly available Swing Controls in Java

jFrame: A Frame is a container control, in which all the controls can be lace.

jLabel: JLable allows placing un-editable text on the Frame/Panel

jTextField: JTextFeild allows placing editable text on the Frame/Pane. User can enter text in a textFiled during runtime.

jbutton: is used to initiate an action when it is clicked.

jList: is a group of values or items from which one or more selections can be made.

jComboBox: jComboBox is similar to jList but also allow to enter editable text during run time. It is a combination of jTextFiled and jList.

jRadioButton: Allow us to choose a single item from a group of jRadioButton options.

jCheckBox: Allow us to choose one or more items from a group of jCheckBox options.

jPasswordField: Allow us to enter a text during the run time but shows an encrypted text instead of the original text

jTextArea: JTextArea is a multi-line text component to enter or edit text.

Focus: The control under execution is said to have the focus. The control having the focus obtains input form the user.

getText(): getText() method is used to obtain the text from a jTextFeild during the run time.

setText(): setText() method is used to set or change the text of a jTextFeild during run time.

Swing Controls Methods and Properties: These are the Swing Controls available with NetBeans IDe and their concern methods and properties are given below.

Swing Controls	Methods	Properties
jButton	• getText() • setText()	Background • Enabled • Font • Foreground • Text • Label
jLabel	• getText()	Background Enabled Font Foreground Text
jTextField	 getText() isEditable() isEnabled() setText() 	 Background Editable Enabled Font Foreground Text
jRadioButton	 getText() setText() isSelected() setSelected() 	Background Button Group Enabled Font Foreground Label Selected
jCheckBox	 getText() setText() isSelected() setSelected() 	Button Group Font Foreground Label Selected Text
jButtonGroup		• Add
jComboBox	•getSelectedItem() •getSelectedIndex() • setModel()	 Background ButtonGroup Editable Enabled Font Foreground Model SelectedIndex SelectedItem Text
jList	• getSelectedValue()	 Background Enabled Font Foreground Model SelectedIndex SelectedItem SelectionMode Text
jTable	addRow()getModel()	• model
JoptionPane	 showMessageDialog() 	 getRowCount() removeRow() addRow()