

RESUME

VENUGOPAL KATUKAM

Email: venu.katukam@gmail.com

Karimnagar.

CAREER OBJECTIVE:

To dedicate myself towards my work and organization and gain valuable experience through working with quality people in a dynamic work environment and strive for collective growth and development by achieving the organizational goals.

EDUCATIONAL QUALIFICATION:

Qualification	Board/University	College	% of Marks	Duration
B.Tech (ECE)	JNTUH	Nigama Engineering college, Karimnagar.	73.6%	2012-16
I.P.E. (MPC)	Board of Intermediate Education, A.P	Trinity Junior College, karimnagar	85%	2010-12
S.S.C.	Board Of Secondary Education, A.P	Siddartha High School, jagtial	90%	2010

SOFTWARE PROFICIENCY

- Programming Languages : C and JAVA
- Tools : Multisim, Masm and Mat Lab.
- Operating systems : Windows and Linux.
- Hardware : Pc total hardware and software
- Data Base : SQL,PLSQL

SKILLS:

- Quick learner, good listener and have good communication skills.
- Fair to deal excellently with my work.

STRENGTHS:

- Self driven personality coupled with problem solving attitude.
- Good communication and Analytical skills.
- Positive attitude, strong will power along with high level of patience.
- Quick learner and good performer both in team and independent job environment.

PROJECT DETAILS

Mini Project: visual navigation alerting system for fisherman.

Major project: Sending sms in an innovative manner using zigbee.

ACHIVEMENTS:

- Event Co-ordinator of NIRMIT'2k13 conducted by Nigama Engineering college, Karimnagar.
- Iam the convener of ABVP srujana national level tech fest in 2015.

PERSONAL DETAILS:

Name	:	VENUGOPAL
Father's Name	:	BHOOMAIAH
Date of Birth	:	18-03-1995
Sex	:	Male
Marital Status	:	Single
Religion	:	Hindu
Languages Known :		Telugu, English, Hindi
Hobbies	:	playing Chess and Listening Music

DECLARATION_:

I here by declare that the above-mentioned information is true to the best of my knowledge.

PLACE:

DATE:

(K.Venugopal)

Core Java Topics & Basic Concepts Complete list

Welcome to your online Java school. Our aim is to deliver a whole training environment for you to become a top class programmer.

Start with lesson 1, or search for a topic manually. Easy lesson plans available for beginners and 2-2 students.

1 Overview Of Programming With Java

1.1 What Is Programming ? Why We Need Programming

1.2 How To Learn Programming & Programming Skills

1.3 How Can I Be A Good/Excellent Programmer

1.4 More details on Java

1.4.1 Platform Independence In Java - WORA & WOCA

1.4.2 Java And Internet - Creating Java Applet

1.4.3 What Is Servlet In Java?

1.4.4 Java Bytecode

1.4.5 Java Buzzwords

1.4.6 JDK JRE JVM JIT - Java Compiler

1.4.7 Java Versions And Changes Done In Every Version

1.4.8 Java Keywords

1.5 Simple Programs and Development environment

1.5.1 Installation Of Java on your PC

1.5.2 Java Sample Program - Simple Hello World Program In Java

1.5.3 How to Compile and Run Java Program In Cmd Prompt

2 Datatypes

2.1 Data Types In Java

2.2 Primitive Datatypes

2.2.1 Primitive Data Types In Java

2.2.2 Integer Data Types In Java

2.2.3 Floating Point Data Types In Java

2.2.4 Java Character

2.2.5 Boolean Data Type In Java

2.2.6 Literals

2.2.6.1 Java Literals

2.2.6.2 Integer Literals In Java

2.2.6.3 Floating Point Literals In Java

2.2.6.4 Character Literals In Java

2.2.6.5 String Literal In Java

2.2.6.6 Boolean Literals In Java

3 Variables

3.1 Java Variables

3.2 Scope Of Variables In Same Block

3.3 Type Conversion In Java

3.4 Type Casting In Java

4 Operators

4.1 Operators In Java

4.2 Java Arithmetic Operators

4.3 Basic Arithmetic Operators In Java

4.4 Increment And Decrement Operators In Java

4.5 Modulus Operator In Java

4.6 Arithmetic Compound Assignment Operators In Java

4.7 Relational Operators In Java

4.8 Boolean Logical Operators In Java

4.9 Short Circuit Logical Operators In Java

4.10 Assignment Operator In Java

4.11 Ternary Operator In Java

4.12 Java Operator Precedence And Associativity

4.13 Temperature Conversion Program In Java

5 Control Statements

5.1 Control Statements In Java

5.2 Selection Statements

5.2.1 Selection Statements In Java

5.2.2 if Condition In Java

5.2.3 Nested if Statements In Java

5.2.4 if else if ladder In Java

5.2.5 switch Statement In Java

- 5.2.6 if else Vs switch Performance In Java**
- 5.2.7 Nested switch Statements In Java**
- 5.2.8 Fall Through Switch Case Statements In Java**
- 5.3 Blocks of code**
 - 5.3.1 Block Of Code In Java**
 - 5.3.2 Scope Of Variables In Nested/Multiple Blocks**
 - 5.3.3 Lifetime Of Variable In Java**
 - 5.3.4 Expressions, Statement, Line & Block In Java**
- 5.4 Iteration statements (Loops)**
 - 5.4.1 Iteration Statements Or Loops In Java**
 - 5.4.2 while Loop In Java**
 - 5.4.3 for Loop In Java**
 - 5.4.4 for Vs while Loop In Java**
 - 5.4.5 do while Loop In Java**
 - 5.4.6 Nested Loops in Java**
 - 5.4.7 Nested While Loop In Java**
 - 5.4.8 Nested for Loop In Java**
 - 5.4.9 for Loop Example Program In Java - Sum Of Numbers**
 - 5.4.10 Factorial Program In Java Using for Loop**
 - 5.4.11 Factorial Program In Java Using While Loop**
- 5.5 Jump Statements**
 - 5.5.1 Jump Statements In Java**
 - 5.5.2 Using Break In for Loop To Exit**
 - 5.5.3 Using break in switch case Statement**
 - 5.5.4 Using Java Break Statements as Java Goto**
 - 5.5.5 Using break In Nested Loop Java Program**
 - 5.5.6 Java continue Statement**
 - 5.5.7 Java return Statement**
- 5.6 Java for loops vs Java while loops vs Java do while loops**
- 6 Methods - Importance**
 - 6.1 Basic Java Methods**
 - 6.2 Java Methods**

- 6.3 Java Methods - Parameter Passing And Scope**
- 6.4 Java Program To Find Simple Interest Using Methods**
- 6.5 Recursive In Java**
- 7 Array - Overview**
- 7.1 Java Array**
- 7.2 Creation And Declaration Of Array In Java**
- 7.3 ArrayList Access Using Index**
- 7.4 Java Multidimensional Array**
- 7.5 Java Array Initialization**
- 7.6 Learn Arrays And Loops**
- 7.7 Java Code To Print Student Details Using Arrays**
- 7.8 For-each Loop In Core Java Programming**
- 7.9 Command Line Arguments In Core Java Programming**
- 8 Classes**
- 8.1 Java Class**
- 8.2 Java Classes and Java Objects**
- 8.3 Java Objects References**
- 8.4 Member Variable In Java**
- 8.5 Class References And Objects In Java**
- 8.6 To Print Student Details Using Classes In Java**
- 8.7 Create Objects Using Constructors In Java**
- 8.8 Class With Multiple Constructors In Java**
- 8.9 this Keyword In Java**
- 8.10 Behavior Of Java Classes Using Methods**
- 8.11 Java Multiple Methods In One Class**
- 8.12 Calling A Class From Another Class In Java**
- 8.13 Creating A Class For Data Validation**
- 8.14 Java Program To Find Rectangle Area & Perimeter Using Classes**
- 8.15 Java Program to Find Area of Various Shapes Using Classes**
- 8.16 Java Program To Compare Movies**
- 9 Class Inheritance**
- 9.1 Java Class Inheritance**

- 9.2 Is-A Relationship In Java**
- 9.3 Passing Sub Class Object As Super Class Reference**
- 9.4 Assigning Sub Class Object To Super Class Reference In Java**
- 9.5 Assigning Super Class Reference To A Sub Class Reference In Java**
- 9.6 Multilevel Inheritance In Java With Example Program**
- 10 Methods Overriding, Overloading**
- 10.1 Method Overloading In Java**
- 10.2 Is Java Pass by Reference or Pass by Value**
- 10.3 Method Overriding In Java**
- 10.4 Inheritance Example Program To Remove Duplicate Code**
- 10.5 How A Method Can Be Overridden In Different Ways**
- 10.6 Method Overloading Vs Method Overriding**
- 10.7 Super Keyword In Java To Call Super Class Constructor**
- 10.8 Inheritance And Constructors In Java**
- 10.9 Dynamic Method Dispatch - Calling Overridden Methods In Java**
- 10.10 Run Time Polymorphism In Java**
- 11 Abstract Class And Methods**
- 11.1 Java Abstract Class**
- 11.2 Abstract Method In Java**
- 11.3 Rules For Abstract Methods and Abstract Classes**
- 11.4 Creating Array Of Objects In Java**
- 11.5 Java Program To Find Largest Area by Comparing Various Shapes**
- 11.6 Java Program For Cricket Players Using Class Hierarchy**
- 12 Interfaces, Packages and Access Control**
- 12.1 Java Interface**
- 12.2 Difference Between Interfaces And Abstract Classes**
- 12.3 Future Task Java Program Using Interfaces**
- 12.4 Creating Interface In Java With Example Program**
- 12.5 Java Package**
- 12.6 How To Compile Classes in Package**
- 12.7 Using private Keyword In Java For Access Control**
- 12.8 Access Modifiers In Java**

- 12.9 Java Access Modifiers With Example Program**
- 13 final, static and others**
 - 13.1 final Keyword In Java**
 - 13.2 Static Keyword In Java**
 - 13.3 Creating Static Methods In Java Using Static Keyword**
 - 13.4 Singleton Design Pattern In Java**
 - 13.5 Java Program To Explain Public Static Void Main**
 - 13.6 Static and Non Static Variables - Static and Non Static Methods**
- 14 Object Oriented Concepts - Revisited**
 - 14.1 Abstraction in Java**
 - 14.2 Polymorphism In Java**
 - 14.3 Encapsulation In Java**
 - 14.4 Inheritance In Java**
- 15 Exceptions**
 - 15.1 Why Java Throws Exceptions**
 - 15.2 How To Handle An Exception In Java**
 - 15.3 Exception Handling In Java with Example Program**
 - 15.4 Try Catch Block In Java**
 - 15.5 Java Multiple Catch Block With Example Program**
 - 15.6 Java Finally Block In Exception Handling**
 - 15.7 User Defined Exception In Java**
 - 15.8 Java Throw Keyword - Java Throws Keyword**
 - 15.9 Difference Between Error and Exception in Java**
 - 15.10 Checked Exception Vs Unchecked Exception In Java**
 - 15.11 Java Built In Exceptions Checked Exceptions, Unchecked Exceptions**
 - 15.12 Exception Handling Syntax In Java Programming**
- 16 Multithreaded Programming**
 - 16.1 Thread Concept In Java**
 - 16.2 The Java Thread Model**
 - 16.2.1 Creation Of Threads In Java**
 - 16.2.2 Java Inter Thread Communication With Example**
 - 16.3 Synchronization**

16.3.1 Thread Synchronization In Java Using 'Synchronized'

16.3.2 static synchronized In Java

16.3.3 Java Synchronized Blocks

16.4 Handling Thread DeadLock In Java

16.5 Java Thread Group

16.6 Modern Ways Of Suspending, Resuming And Stopping Threads In Java

17 Generics

17.1 Java Generics

17.2 A Simple Generics Example

17.2.1 How Generics Improve Type Safety In Java

17.3 A Generic Class With Two Type Parameters In Java

17.4 Java Bounded Type - Bounded Type In Java

17.5 Generics Wildcards In Java With Examples

17.6 Java Generics In Methods And Constructors

17.7 Generic Interface In Java

17.8 Java Type Erasure

18 Strings

18.1 Java String

18.2 Java length() Method | length() Method In Java - Strings

18.3 Special String Operations

18.3.1 Literals In Java

18.3.2 Java String concatenation - concat() Method In Java

18.3.3 Java String Concatenation with Other Data Types

18.3.4 Java String Conversion - toString() Method In Java

18.4 Character Extraction

18.4.1 charAt() Method In Java - Java Character Extraction

18.4.2 getChars() Method In Java

18.4.3 Java Character Extraction - Java String getBytes() Method

18.4.4 Java Character Extraction - toCharArray() Method In Java

18.5 String Comparison

18.5.1 Java String Comparison Methods - Equals and EqualsIgnoreCase

18.5.2 Java regionMatches() Method - String Comparison

18.5.3 Java String startsWith() And endsWith() Methods

18.5.4 Java equals method vs == Operator

18.5.5 Java compareTo() method

18.6 Java Searching Strings - Java indexOf, lastIndexOf Methods

18.7 Modifying a String

18.7.1 Java String substring() method - substring In Java

18.7.2 concat() method In Java

18.7.3 replace() Method In Java

18.7.4 Java String trim() Method - trim() Method In Java

18.8 Data Conversion Using valueOf In Java

18.9 toLowerCase() And toUpperCase() Methods In Java

18.10 Additional String Methods in Java

18.11 Java String Arrays - String Arrays In Java

18.12 StringBuffer

18.12.1 Java StringBuffer

18.12.2 Java StringBuffer

18.12.3 Java StringBuffer length() And capacity() Methods

18.12.4 Java StringBuffer ensureCapacity() Method With Example

18.12.5 Java setLength() Method In StringBuffer Class

18.12.6 Java charAt() And setCharAt() Methods in StringBuffer

18.12.7 StringBuffer getChars() Method In Java With Example

18.12.8 Java append() Method In StringBuffer

18.12.9 Java StringBuffer insert() Method With Example

18.12.10 Java StringBuffer, reverse() - Reverse A String In Java

18.12.11 Java delete() and deleteCharAt() Methods In StringBuffer

18.12.12 Java StringBuffer replace() Method With Example

18.12.13 Java substring

18.12.14 Additional StringBuffer Methods In Java

18.13 StringBuilder Class In Java

18.14 Conclusion Of Strings In Java

19 Exploring java.lang

19.1 Primitive Type Wrappers

19.1.1 Java Number Class

19.1.2 Java Double Class And Java Float Class

19.1.3 Java isInfinite() And isNaN() Methods In Double Class

19.1.4 Creating Objects for Primitive Data Types (Byte, Short)

19.1.5 Converting Numbers to and from Strings In Java

19.1.6 Java Character Class

19.1.7 Character Unicode, Code Point Support In Java

19.1.8 Java Boolean Class

19.2 Java Void Class

19.3 Java Process Class

19.4 Java Runtime Class - java.lang.Runtime

19.5 Java ProcessBuilder

19.6 System

19.6.1 Using currentTimeMillis() Method In Java

19.6.2 System Class arraycopy() Method In Java

19.6.3 Java Environment Properties

19.7 Object As A Super Class In Java

19.8 clone() Method And cloneable Interface In Java

19.9 java.lang.class - Java Library

19.10 Java ClassLoader

19.11 Java Math Class - java.lang.Math

19.12 Java Package Class

19.13 Java Enumeration

19.14 Java Comparable Interface with Example

19.15 Conclusion (Exploring java.lang)

20 Collections Framework

20.1 Java Collections Overview

20.2 Collection Interface

20.2.1 Java List Interface

20.2.2 Set Interface In Java

20.2.3 Java SortedSet Interface

20.2.4 Java NavigableSet Interface

20.3 Collection Classes

20.3.1 Java ArrayList

20.3.2 Java LinkedList

20.3.3 HashSet Class In Java

20.3.4 Java LinkedHashSet

20.3.5 Java TreeSet - TreeSet Examples in Java

20.3.6 Java PriorityQueue - PriorityQueue In Java

20.3.7 Java ArrayDeque Class

20.3.8 Java EnumSet

20.4 Iterator

20.4.1 Java Iterator

20.4.2 List Iterator In Java

20.5 Map Interfaces

20.5.1 Java Map Interfaces - HashMap, TreeMap, LinkedHashMap

20.5.2 Java SortedMap Interface

20.5.3 Java NavigableMap

20.5.4 Java Map.Entry Interface

20.6 Map Classes

20.6.1 Java HashMap Implementation

20.6.2 TreeMap In Java - java.util.TreeMap

20.6.3 Java WeakHashMap Class

20.6.4 LinkedHashMap In Java with Code Example

20.6.5 Java IdentityHashMap

20.6.6 Java EnumMap

20.7 Java Comparators

20.8 Collection Algorithms

20.8.1 Java Collection Algorithms

20.8.2 Java Read-only Collections And Algorithms

20.8.3 Java Thread Safe Collections & Algorithms

20.8.4 Java Singleton

20.8.5 Java nCopies Collections - Collections.nCopies() Method

20.9 java.util.Arrays - Class Arrays In Collection Framework

20.10 Why Collections Are Generic In Java?

20.11 Legacy Classes and Interfaces

20.11.1 Java Enumeration Interfaces - Java Enumeration Examples

20.11.2 Java Vector

20.11.3 Stack In Java - java.util.Stack Class

20.11.4 Java Dictionary Class - java.util.Dictionary

20.11.5 Java Hashtable

20.11.6 Java Properties Class - java.util.Properties Class

20.12 Collection Framework In Java

21 More Utility Classes

21.1 Java Collections - Utility Classes In Java

21.2 StringTokenizer In Java

21.3 Java BitSet

21.4 Java Date Class

21.5 Calendar In Java - java.util.Calendar Class

21.6 Java GregorianCalendar

21.7 Java TimeZone Class

21.8 Java SimpleTimeZone

21.9 Locale Class In Java

21.10 Java Random Class - java.util.Random Package

21.11 Java Observable

21.12 Java Timer Class And Java TimerTask Class

21.13 Java Currency Class

21.14 Formatter

21.14.1 Java Formatter Class

21.14.2 Java Formatter Methods

21.14.3 Java Formatter

21.14.4 Formatting Strings And Characters By Using Formatter

21.14.5 Java Formatting Numbers

21.14.6 Formatting Date And Time In Java With Example

21.15 Scanner

- 21.15.1 Java Scanner Class Constructors With Example**
- 21.15.2 Java Scanner Class Methods With Examples**
- 21.16 Java ResourceBundle, ListResourceBundle And PropertyResourceBundle Classes**
- 21.17 java.util Subpackages**
 - 21.17.1 Usage Of java.util.logging Package**
 - 21.17.2 Java Regular Expression**
- 22 Input/Output: Exploring java.io**
 - 22.1 Java Input/Output Classes And Interfaces**
 - 22.2 File**
 - 22.2.1 Java Directories - isDirectory() Method In Java**
 - 22.2.2 Using FilenameFilter Interface In Java**
 - 22.2.3 Alternative For list() Method - listFiles() Method**
 - 22.2.4 Creating Directories In Java - Creating Java Directories**
 - 22.3 AutoCloseable, Closeable And Flushable Interfaces In Java**
 - 22.4 Java I/O Exceptions - I/O Exceptions In Java**
 - 22.5 Two Ways To Close A Stream In Java**
 - 22.6 Java Stream Classes**
 - 22.7 Byte Streams**
 - 22.7.1 Java InputStream Class**
 - 22.7.2 Java OutputStream Class**
 - 22.7.3 Java FileInputStream**
 - 22.7.4 Java FileOutputStream**
 - 22.7.5 Java ByteArrayInputStream**
 - 22.7.6 Java ByteArrayOutputStream**
 - 22.7.7 Java Filtered Byte Streams**
 - 22.7.8 Java Buffered Byte Streams**
 - 22.7.9 Java BufferedInputStream**
 - 22.7.10 Java BufferedOutputStream - BufferedOutputStream In Java**
 - 22.7.11 Java PrintStream Class**
 - 22.7.12 DataInputStream And DataOutputStream In Java**
 - 22.7.13 Java RandomAccessFile**

22.8 Character Streams

22.8.1 Reader Class In Java

22.8.2 Java Writer Class

22.8.3 Java FileReader

22.8.4 Java FileWriter

22.8.5 Java CharArrayReader

22.8.6 Java CharArrayWriter

22.8.7 BufferedReader In Java

22.8.8 Java BufferedWriter

22.8.9 PrintWriter Class In Java

22.9 Java Console Class

22.10 Serialization

22.10.1 Serialization In Java

22.10.2 Externalizable In Java with Example

22.10.3 Java ObjectOutputStream

22.10.4 Java ObjectOutputStream

22.10.5 Java ObjectInput

22.10.6 Java ObjectInputStream

22.10.7 Java Serialization Process -

22.11 Java Stream Benefits

22.12 Conclusion To Input/Output (Exploring java.io)

23 Other Core Java Topics

23.1 Other Advanced Topics In Java

23.2 Advanced Java Programming Concepts