ACADEMIC PLANNER

For

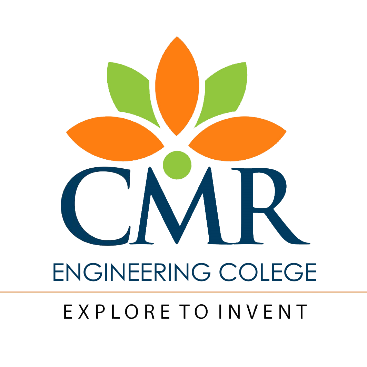
***“*Object Oriented Programming through Java*”***

**Presented by**

**Radhe Shyam Panda**

Department of

**CSE(AI&ML)**

****

**CMR ENGINEERING COLLEGE**

(Approved by AICTE-NewDelhi, Affiliated to J.N.T.U, Hyderabad)

Kandlakoya(v),Medchal Road,Hyderabad-501 401,Telangana State, India .Website: www.cmrec.ac.in

**(AY:2025-26)**

**ACADEMIC PLANNER**

**Subject: Object Oriented Programming through Java**

**S.NO CONTENT**

**(1) - Preamble/Introduction**

**(2) - Prerequisites**

**(3) - Objectives and Outcomes**

**(4) - Syllabus**

**1.R22-CMREC**

**2.GATE**

**3.IES**

**(5) - List of Expert Details**(Local/National/International with Contact details/Profile link/Blogs/their research Contribution towards the subject)

**(6) - Journals with min 5 ref paper for literature study**

**(7) - Subject -Lesson plan**

**(8) - Suggested Books (**Prescribed and References)

**(9) - Websites for self-learning** **resources like**

*www.geeksforgeeks.org, www.schools.com,* [*Coursera*](https://www.theeducationmagazine.com/word-art/best-educational-websites/#Coursera)*,* [*edX*](https://www.theeducationmagazine.com/word-art/best-educational-websites/#edX)*,* [*Udemy*](https://www.theeducationmagazine.com/word-art/best-educational-websites/#Udemy)*,* [*Khan Academy*](https://www.theeducationmagazine.com/word-art/best-educational-websites/#Khan Academy), NPTEL etc along Registration procedures*)*

**(10) - Question Banks 1.JNTUH/CMREC/Model papers**

**2.GATE**

**(11) - Two case study presentations with Project / Product/ Model /prototypes/ Industrial applications. (12) - Assignment Question/Innovative Assignments sets.**

**(13) - List of topics for students Seminars with Guidelines**

**(14) - STEP/Course material in softcopy**

**(15) - Expert Lectures with topics &Schedules (if any)**

1. **Preamble/Introduction:**

Java is an object oriented programming language developed at Sun Microsystems inc by James Gosling in 1991, The first publicly available version of Java (Java 1.0) was released in 1995.Later on Java got acquired by Oracle Corporation. It is an object oriented and platform-independent programming language because in Java, programs are compiled into byte code and that byte code is platform-independent. The byte code is executed by the Java Virtual Machine

**2.Prerequisites**

Some knowledge of basic programming concepts would help the student to understand it .

**3. OBJECTIVES & OUTCOMES**

1. **Objectives**

* To be able to **understand** the basic object-oriented programming concepts and apply these in problem solving.
* To be able to **demonstrate** inheritance concepts for reusing the program.
* To be able to **design** multitasking by using multiple threads and event handling
* To be able to **understand** event handling.
* To be able to **design** the basics of java console and GUI based programming

1. **Outcomes**

* Able to solve real world problems using OOP techniques.
* Able to understand the use of abstract classes.
* Able to solve problems using java collection framework and I/O classes.
* Able to develop multithreaded applications with synchronization.
* Able to develop applets for web applications and GUI application.

**4. SYLLABUS**

**4.1 CMREC Autonomous Syllabus**

**Unit – 1**

**Object oriented thinking and Java Basics**- Need for OOP paradigm, summary of OOP concepts, coping with complexity, abstraction mechanisms. A way of viewing world – Agents, responsibility, messages, methods, History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, method binding, inheritance, overriding and exceptions, parameter passing, recursion, nested and inner classes, exploring string class.

**Unit – 2**

**Inheritance, Packages and Interfaces**–Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

**Unit – 3**

**Exception handling and Multithreading**—Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. String handling, exploring java.util. Differences between multithreading and multitasking, thread lifecycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads. Enumerations, auto boxing, annotations, generics.

**Unit – 4**

**Event Handling:** Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components-labels, button, canvas, scroll bars, text components, checkbox, check box groups, choices, lists panels– scroll pane, dialogs, menu bar, graphics, layout manager –layout manager types–border, grid, flow, card and grid bag.

**Unit - 5**

**Applets** – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing-JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons–The JButton class, Checkboxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees and Tables.

**4.2 GATE Syllabus**

Basics of Programming – Recursion, Array, Strings, Exception Handling, Collection Framework.

**4.3 IES**

NA

1. **Expert Details**
   * **International**

**James Gosling**

Engineer at Amazon Web Services (AWS) and the original developer of Java, Blog: [**nighthacks.com/roller/jag**](http://nighthacks.com/roller/jag)

* + **National**

**Mr. Ranganath Vadapalli,**

Java full stack developer, JP Morgan Chase, Email - [vadapalliranganath@gmail.com](mailto:vadapalliranganath@gmail.com),

Contact No -8884057329, Expert in Java, java8, Spring Boot, Angular and React JS.

**Mr**. **Shubham Giri**

Software Architect , Amdocs,Pune

Contact No-7869521631

* + **Local**

**Dr. Vignesh Janarthanan**

Professor, HOD, Department of CSE, Malla Reddy Institute of Technology and Science

Contact No-8106193443, Email id: vigneshj2004@gmail.com

**6. JOURNALS**

* 1. <https://ieeexplore.ieee.org/document/9712153>
  2. [https](Design%20of%20Multithreaded%20Software%20-%202011%20-%20Sand%20n%20-%20Frontmatter.pdf)://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470904916.fmatter
  3. [An interactive environment for beginning Java programmers](https://www.sciencedirect.com/science/article/pii/S0167642304000590)
  4. [PAMELA: An annotation-based Java modeling framework - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0167642321000617)
  5. [A type-directed algorithm to generate random well-typed Java 8 programs - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0167642320301039)
  6. [Concept–based Analysis of Java Programming Errors among Low, Average and High Achieving Novice Programmers](https://www.informingscience.org/Publications/4322?Source=%2FJournals%2FJITEIIP%2FArticles%3FVolume%3D0-0)
  7. [Java Tutorial, Java EE Tutorials - JournalDev](https://www.journaldev.com/java-tutorial-java-ee-tutorials)

**7. Subject Lesson Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Topic (syllabus)** | **Sub-Topic** | **NO. OF LECTURES REQUIRED** | **Suggested Books** | **Teaching Methods** |
| 1 | **Unit – 1** |  |  |  |  |
| 2 | **Object oriented thinking and Java Basics Thinking** | Agents and Communities, messages and methods, and Responsibilities | 1 | T1, T2 | BB / PPT |
| 3 |  | Classes and Instances, Class Hierarchies |  | T1, T2 | BB / PPT |
| 4 |  | Inheritance, Method binding | 1 | T1, T2 | BB / PPT |
| 5 |  | Overriding and Exceptions |  | T1, T2 | BB / PPT |
| 6 |  | Summary of Object-Oriented concepts | 1 | T1, T2 | BB / PPT |
| 7 |  | Java buzzwords | 1 | T1, T2 | BB / PPT |
| 8 |  | An Overview of Java |  | T1, T2 | BB / PPT |
| 9 |  | Data types, Variables and Arrays | 1 | T1, T2 | BB / PPT |
| 10 |  | operators, expressions | 1 | T1, T2 | BB / PPT |
| 11 |  | control statements | 1 | T1, T2 | BB / PPT |
| 12 |  | Introducing classes, Methods and Classes | 1 | T1, T2 | BB / PPT |
| 13 |  | String handling | 1 | T1, T2 | BB / PPT |
| 14 |  | Inheritance concept, Inheritance basics | 1 | T1, T2 | BB / PPT |
| 15 |  | Member access, Constructors | 1 | T1, T2 | BB / PPT |
| 16 |  | Creating Multilevel hierarchy | 1 | T1, T2 | BB / PPT |
| 17 |  | super uses, using final with inheritance | 1 | T1, T2 | BB / PPT |
| 18 |  | Polymorphism-ad hoc polymorphism | 1 | T1, T2 | BB / PPT |
| 19 |  | pure polymorphism | 1 | T1, T2 | BB / PPT |
| 20 |  | method overriding, abstract classes, Object class | 1 | T1, T2 | BB / PPT |
| 21 |  | forms of inheritance- specialization, specification, construction, extension, limitation, combination | 1 | T1, T2,R1 | BB / PPT |
| 22 |  | benefits of inheritance and costs of inheritance | T1, T2 | BB / PPT |
| 23 |  | **Total classes required for UNIT - 1** | 17 |  |  |
| 24 | **UNIT – 2** |  |  |  |  |
| 25 | **Inheritance, Packages and Interfaces** | Defining a Package | 1 | T1, T2 | BB / PPT |
| 26 |  | CLASSPATH | T1, T2 | BB / PPT |
| 27 |  | Access protection | T1, T2,R3 | BB / PPT |
| 28 |  | importing packages | 1 | T1, T2 | BB / PPT |
| 29 |  | defining an interface | T1, T2 | BB / PPT |
| 30 |  | implementing interfaces | 1 | T1, T2 | BB / PPT |
| 31 |  | Nested interfaces | T1, T2 | BB / PPT |
| 32 |  | applying interfaces | 1 | T1, T2,R4 | BB / PPT |
| 33 |  | variables in interfaces | T1, T2 | BB / PPT |
| 34 |  | extending interfaces | T1, T2 | BB / PPT |
| 35 |  | classes-Byte streams and Character streams | 1 | T1, T2 | BB / PPT |
| 36 |  | Reading console Input and Writing Console Output | T1, T2 | BB / PPT |
| 37 |  | File class | 1 | T1, T2 | BB / PPT |
| 38 |  | Reading and writing Files | T1, T2 | BB / PPT |
| 39 |  | Random access file operations | 1 | T1, T2 | BB / PPT |
| 40 |  | The Console class | T1, T2 | BB / PPT |
| 41 |  | Serialization | 1 | T1, T2 | BB / PPT |
| 42 |  | Enumerations | 1 | T1, T2 | BB / PPT |
| 43 |  | auto boxing | T1, T2 | BB / PPT |
| 44 |  | Generics | T1, T2 | BB / PPT |
| 45 |  | **Total classes required for UNIT - 2** | 9 |  |  |
| 46 | **UNIT – 3** |  |  |  |  |
| 47 | **Exception handling and Multithreading** | Fundamentals of exception handling | 1 | T1, T2 | BB / PPT |
| 48 |  | Exception types | T1, T2 | BB / PPT |
| 49 |  | Termination or resumptive models | 1 | T1, T2 | BB / PPT |
| 50 |  | Uncaught exceptions | T1, T2 | BB / PPT |
| 51 |  | using try and catch | 1 | T1, T2 | BB / PPT |
| 52 |  | multiple catch clauses | T1, T2 | BB / PPT |
| 53 |  | nested try statements | 1 | T1, T2 | BB / PPT |
| 54 |  | throw, throws and finally | T1, T2 | BB / PPT |
| 55 |  | built- in exceptions | 1 | T1, T2 | BB / PPT |
| 56 |  | creating own exception sub classes | 1 | T1, T2 | BB / PPT |
| 57 |  | Differences between thread-based multitasking and process-based multitasking | 1 | T1, T2 | BB / PPT |
| 58 |  | Java thread model | T1, T2 | BB / PPT |
| 59 |  | creating threads | 1 | T1, T2 | BB / PPT |
| 60 |  | thread priorities | 1 | T1, T2 | BB / PPT |
| 61 |  | synchronizing threads | T1, T2 | BB / PPT |
| 62 |  | inter thread communication | 1 | T1, T2 | BB / PPT |
| 63 |  | **Total classes required for UNIT - 3** | 10 |  |  |
| 64 | **UNIT – 4** |  |  |  |  |
| 65 | **Event Handling:** | Collections overview | 1 | T1, T2 | BB / PPT |
| 66 |  | Collection Interfaces | T1, T2 | BB / PPT |
| 67 |  | The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque | 2 | T1, T2 | BB / PPT |
| 68 |  | Accessing a Collection via an Iterator | 1 | T1, T2 | BB / PPT |
| 69 |  | Using an Iterator | T1, T2 | BB / PPT |
| 70 |  | The For-Each alternative | T1, T2 | BB / PPT |
| 71 |  | Map Interfaces and Classes | 1 | T1, T2 | BB / PPT |
| 72 |  | Comparators | T1, T2 | BB / PPT |
| 73 |  | Collection algorithms | T1, T2 | BB / PPT |
| 74 |  | Arrays | T1, T2 | BB / PPT |
| 75 |  | The Legacy Classes and Interfaces- Dictionary, Hashtable ,Properties, Stack, Vector More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner | 3 | T1, T2 | BB / PPT |
| 76 |  | **Total classes required for UNIT - 4** | 8 |  |  |
| 77 | **UNIT – 5** |  |  |  |  |
| 78 | **Applets** | Introduction and limitations of Applet | 1 | T1, T2,R2 | BB / PPT |
| 79 |  | MVC architecture through AWT | T1, T2 | BB / PPT |
| 80 |  | Components | 1 | T1, T2 | BB / PPT |
| 81 |  | Containers | T1, T2 | BB / PPT |
| 82 |  | Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout | 1 | T1, T2 | BB / PPT |
| 83 | **Event Handling** | The Delegation event model | 1 | T1, T2 | BB / PPT |
| 84 |  | Events, Event sources, Event Listeners, and Event classes | T1, T2 | BB / PPT |
| 85 |  | Handling mouse and keyboard events | 1 | T1, T2,R3 | BB / PPT |
| 86 |  | Adapter classes | 1 | T1, T2 | BB / PPT |
| 87 |  | Inner classes | T1, T2 | BB / PPT |
| 88 |  | Anonymous Inner classes | T1, T2 | BB / PPT |
| 89 |  | Applets and HTML | 1 | T1, T2 | BB / PPT |
| 90 |  | Security Issues | T1, T2 | BB / PPT |
| 91 |  | Applets and Applications | T1, T2 | BB / PPT |
| 92 |  | passing parameters to applets | 1 | T1, T2 | BB / PPT |
| 93 |  | Creating a Swing Applet | T1, T2 | BB / PPT |
| 94 |  | Painting in Swing | T1, T2,R5 | BB / PPT |
| 95 |  | A Paint example | 1 | T1, T2 | BB / PPT |
| 96 |  | Exploring Swing Controls- JLabel and Image Icon, JText Field | T1, T2 | BB / PPT |
| 97 |  | JButton | 2 | T1, T2 | BB / PPT |
| 98 |  | JToggle Button | T1, T2 | BB / PPT |
| 99 |  | JCheck Box | T1, T2 | BB / PPT |
| 100 |  | JRadio Button | T1, T2 | BB / PPT |
| 101 |  | JTabbed Pane | T1, T2 | BB / PPT |
| 102 |  | JScroll Pane | T1, T2 | BB / PPT |
| 103 |  | JList | T1, T2 | BB / PPT |
| 104 |  | JCombo Box | T1, T2 | BB / PPT |
| 105 |  | Swing Menus | T1, T2 | BB / PPT |
| 106 |  | Dialogs | T1, T2 | BB / PPT |
| 107 |  | **Total classes required for UNIT - 5** | 11 |  |  |
| 108 |  | **Total classes required** | **55** |  |  |

**8. Suggested Books**

**Text Books**

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.[T1]
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education. [T2]

**Reference Books**

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons [R1]
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education. [R2]
3. Object Oriented Programming through Java, P. Radha Krishna, University Press. [R3]
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press. [R4]
5. Java Programming and Object-oriented Application Development, R. A. Johnson, Cengage Learning. [R5]
6. Core and Advanced Java, Black Book, Steven Holzner [R5]

**9.WEBSITES For Self Learning Resources**

* 1. <https://nptel.ac.in/courses/106/105/106105191/>
  2. <http://www.btechsmartclass.com/java/java-tutorials.html>
  3. <https://www.tutorialspoint.com/java/index.htm>
  4. <https://www.w3schools.com/java/default.asp>
  5. <https://www.programiz.com/java-programming>
  6. <https://www.guru99.com/java-tutorial.html>

**10. Question Banks**

****

**Short Answer Question**

1. Java is platform-independent and portable. - Justify
2. Java is distributed language – Justify
3. Java is dynamic and extensible language - Justify
4. What is JVM (Java Virtual Machine)?
5. List out Java development tools and explain any one from it
6. List out Java API and define any one of it
7. Draw the Java program structure
8. Differences between print() and println() in java.
9. Write down the rules for variable naming.
10. What are symbolic constants? Explain with example.
11. What is meaning of automatic type conversion?
12. List out the types of operators used in Java
13. How can you use ternary operator?
14. Define the break and continue statements.
15. List out the features of Java.
16. List the primitive and non-primitive data types used in Java.
17. Write the syntax of any one of the following loops – while, do...while, for loop.
18. Write various types of inheritance.
19. Define method overloading.
20. Define fields and methods of a class in java.
21. Define constructor. How do we invoke constructor in java?
22. Define method overriding.
23. Define Inheritance. How it is created in Java?
24. What is the use of auto boxing in java? Explain.
25. Define interface. Write the syntax for implementing an interface in a class.
26. What is the meaning of the *static* keyword?
27. What are the restrictions with static methods?
28. Differentiate between Classes and Interface.
29. Define the final variable and final methods.
30. Define class. How does it accomplish data hiding?
31. What is the meaning of the *abstract* keyword?
32. What are the conditions for using super() method.
33. Give examples of the Run-time error.
34. List out the different types of exception.
35. Define the terms: try, catch.
36. Define the term - stream, reader stream classes, writer stream classes.
37. Explain any two string methods.
38. List the java API packages.
39. List the methods of Reader or Writer class.
40. Define the term – Exception and Exception Handling.
41. What is event delegation?
42. List out methods of MouseListener.
43. Define (1) Event (2) Event Source (3) Event Class (4) Event Listener
44. Explain methods (1) setBackground( ) (2) setForeground( )
45. Write difference between java applet program and java application program.
46. List down methods for KeyEvent class and ItemEvent class.
47. What is synchronization and why is it important?
48. What is the purpose of String Tokenizer class? Explain.
49. What is an adapter class? Explain with an example.
50. What is a Collection Class? Give an example.

**Long Answer Question**

1. What is meant by byte code? Briefly explain how java is platform independent.
2. Explain the significance of public, private, and protected access specifiers in inheritance.
3. Explain different parts of a java program with an appropriate example.
4. How does polymorphism promote extensibility? Explain with example.
5. Explain the process of defining and creating package with suitable example.
6. Give an example where interface can be used to support multiple inheritance.
7. What is the accessibility of public method or field inside a non-public class or interface? Explain.
8. Describe the process of importing and accessing a package with suitable examples.
9. Differentiate between checked and unchecked exceptions with examples.
10. Write a program to create four threads using Runnable interface.
11. What are different ways to handle exceptions? Explain.
12. How many ways are possible in java to create multiple threaded programs? Discuss the difference between them.
13. Differentiate between ArrayList and Vector? Why ArrayList is faster than Vector? Explain.
14. How a Hashtable can change the iterator? Explain.
15. Explain the Bit Set and Calendar classes in detail?
16. Discuss the differences between HashList and HashMap, Set and List?
17. List and explain different types of Layout managers with suitable examples.
18. How to move/drag a component placed in a swing container? Explain.
19. Discuss about different applet display methods in brief.
20. What are the various components of a swing? Explain.
21. Write the significance of JVM.
22. How to implement polymorphism in java? Explain briefly.
23. What is an array? How do you declare an array in java? Give examples.
24. How to design and implement interface in java? Give an example.
25. Write about the methods available in the Character Streams? Discuss.
26. Distinguish between Byte Stream class and Character stream classes.
27. What is an exception? How the exceptions are handled in java?
28. Write a java program that illustrate the application of multiple catch statements.
29. Differentiate between multiprocessing and multi-threading. What is to be done to implement these in a Program?
30. Write a program that creates two threads. First thread prints numbers from 1 to 100 and other thread prints numbers from 100 to 1.
31. What are the similarities between ArrayList and Vector? Explain.
32. What is the difference between Iterator and ListIterator? Explain different ways to iterate over a list.
33. What is Comparable and Comparator interface? Differentiate between them.
34. What is the difference between init() and start() methods in Applet? When will each be executed?
35. Create an applet to draw a Cube and Circle shapes.
36. Write a program to create a frame window that responds to mount clicks.
37. Describe different levels of access protection available in java.
38. List the primitive data types available in java and explain.
39. How can you extend one interface by the other interface? Discuss.
40. Discuss about CLASSPATH environmental variable.
41. Write a java program that demonstrates how certain exception types are not allowed to be throw.
42. What is the importance of hashCode() and equals() method.
43. What is an applet? Explain the life cycle of an applet with a neat sketch.
44. Write a program to create a frame window that responds to key strokes.

**11.CASE STUDY**

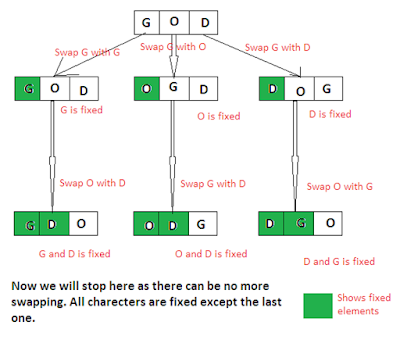
**11.1 Avoiding deadlock in Java (**[**Solution**](deadlock_solution.docx)**)**

This is one of the interesting programs from Java Interviews, mostly asked write code where a resource is accessed by multiple threads. You need to write code in such a way that no deadlock should occur. The trick to solving this problem is acquiring resources in order and release them in reverse order, e.g. first acquire resource R1 and only if you have got R1 to go for R2. This way, you can avoid deadlock.

**11.2 Print all permutations of String (**[**Solution**](String%20Permutation.docx))

Write a Java program to print all permutations of a given String. For example, if given tring is "GOD" then your program should print all 6 permutations of this string,

e.g. "GOD," "OGD," "DOG," "GDO," "ODG," and "DGO."

[](http://javarevisited.blogspot.sg/2015/08/how-to-find-all-permutations-of-string-java-example.html)

**12.Assignment and Innovative Assignment Questions**

* **Assignment – 1**

1. What is meant by byte code? Briefly explain how Java is platform independent?
2. Explain the significance of public, protected and private access specifiers in inheritance.
3. Write the significance of Java Virtual Machine.
4. How do we implement polymorphism in JAVA? Explain briefly.
5. How to design and implement an interface in java? Give an example.

* **Assignment – 2**

1. What is an Exception? How is an Exception handled in Java?
2. Differentiate between multiprocessing and multi-threading. What is to be done to implement these in a Program?
3. Explain the process of defining and creating a package with suitable examples.
4. What is Java Collections Framework? List out some benefits of Collections framework and explain.
5. What is an applet? Explain the life cycle of Applet with a neat sketch.

**important Question Sets of each Unit**

**Unit – 1**

1. What is Java? Explain the features of Java.
2. Describe the Java environment
3. Explain the structure of Java program
4. Explain the data types available in Java
5. Explain type casting with example
6. Explain the scope of variable
7. List out the decision making statements available in Java. Explain with example
8. List out the looping statements available in Java. Explain with example
9. Write various types of inheritance
10. Define inheritance. Describe different forms of inheritance.

**Unit – 2**

1. Explain the process of defining and creating package with suitable example.
2. Describe the various forms of implementing interface. Give an example of JAVA code for each case.
3. When do we declare a method or class abstract? Discuss with one Example.
4. Write short note on method overloading and method overriding.
5. Define the term - stream, reader stream classes, writer stream classes.

**Unit – 3**

1. Give examples of the Run-time error
2. Explain Arithmetic Exception with an example.
3. What is exception? Explain the syntax of try block and catch block with an example.
4. Describe the try and catch statements in detail.
5. Differentiate between multiprocessing and multi-threading. What is to be done to implement these in a Program?
6. Write a program that creates two threads. First thread prints numbers from 1 to 100 and other thread prints numbers from 100 to 1.

**Unit – 4**

1. Discuss the differences between HashList and HashMap, Set and List?
2. What are the similarities between ArrayList and Vector? Explain.
3. What is the difference between Iterator and ListIterator? Explain different ways to iterate over a list.
4. What is Comparable and Comparator interface? Differentiate between them.

**Unit – 5**

1. Explain Applet life cycle in detail.
2. Write short note on following components.
   1. Label
   2. TextField
   3. TextArea
   4. List
   5. Choice
   6. Button
   7. Checkbox
3. Define (1) Event (2) Event Source (3) Event Class (4) Event Listener

|  |
| --- |
| 1. Write difference between java applet program and java application program. |
| 1. List down methods for KeyEvent class and ItemEvent class |

**13. List of Topics for Students Seminar**

* Inheritance in java
* Exception handling in java
* Interfaces in java
* Packages in java
* Collections in java
* AWT and Swings

**14.** [**STEP MATERIAL**](file:///C:\Users\Azhar\Downloads\JAVA%20PROGRAMMING_STEP_MATERIAL.docx)—



**15. EXPERT LECTURE SCHEDULE:**

Real time applications of multi-threading and swing applications – Tentative period in month of November First week 2025.