

SAVE TREES USE E-RESOURCES

**ALL COURSES | ALL SEMESTERS | NOTES |
SOLVED QUESTION PAPER | LAB MANUALS**



BU **STUDYMATE**

Connecting Minds...!



/Bustudymate.in



70194 28102

Click here to Join Our WhatsApp Group



Soundarya Educational Trust

Soundarya Institute of Management & Science
Affiliated to Bangalore University || NAAC Accredited Institution || Approved by Govt.
of Karnataka

Department of Computer Science



BCA 504T JAVA PROGRAMMING



BCA504T: OBJECT ORIENTED PROGRAMMING USING JAVA

Total Teaching Hours: 60

No of Hours / Week: 04

Unit - I

Introduction to JAVA: JAVA Evolution: Java History, Java Features, How Java Differs from C and C++, Java and Internet, Java and World Wide Web, Web Browsers, Hardware and Software Requirements, Java Support Systems, Java Environment. Overview of JAVA Language: Introduction, Simple Java program, More of Java Statements, Implementing a Java Program, Java Virtual Machine, Command Line Arguments, Programming Style. Constants, Variables, and Data Types: Introduction, Constants, Variables, Data Types, Declaration of Variables, Giving Values to Variables, Scope of Variables, Symbolic Constants, Type Casting, Getting Values of Variables, Standard Default Values, Operators and Expressions: Introduction, Arithmetic Operators, Relational Operators Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Bitwise Operators, Special Operators, Arithmetic Expressions, Evaluation of Expressions, Precedence of Arithmetic Operators, Type Conversion and Associativity, Mathematical Functions. Decision Making and Branching: Introduction, Decision Making with if Statement, Simple if Statement, The if.....else Statement, Nesting of if.....Else Statements, The else if Ladder, The Switch Statement, The ?: Operator. Decision Making and Looping: Introduction. The while Statement, The do Statement, The for Statement, Jumps in Loops Labeled Loops.

[12 hours]

Unit - II

Classes, Arrays, Strings and Vectors: Classes, Objects and Methods: Introduction, Defining a Class, Adding Variables, Adding Methods, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members, Nesting of Methods, Inheritance: Extending a Class Overriding Methods, Final Variables and Methods, Finalizer methods, Abstract Methods and Classes, Visibility Control. Arrays, Strings and Vectors: Arrays, One-dimensional Arrays, Creating an Array, Two -Dimensional Arrays, Creating an Array, Two – dimensional Arrays, Strings, Vectors, Wrapper Classes.

[12 Hours]

Unit - III

Interfaces, Packages, and Multithreaded Programming: Interfaces: Multiple Inheritance: Introduction, Defining Interfaces, Extending Interfaces, Implementing Interfaces, Accessing Interface Variables. Packages: Putting Classes together: Introduction, Java API Packages, Using System Packages, Naming Conventions, Creating Packages, Accessing a Package, Using a Package, Adding a Class to a Package, Hiding Classes. Multithreaded Programming: Introduction, Creating Threads, Extending the Thread Class, Stopping and Blocking a thread, Life Cycle of a thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the ‘Runnable’ Interface. [12 Hours]

Unit - IV

Managing Exceptions, Applet Programming: Managing Errors and Exception: Introduction, Types of Exception Handling Code, Multiple Catch Statements, Using Finally Statement, Throwing Our Own Exceptions, Using Exceptions for Debugging. Applet Programming: Introduction, How Applets

Differ from Applications, Preparing to Write Applets, Building Applet Code, Applet Life Cycle, Creating an Executable applet, Designing a Web Page, Applet Tag, Adding Applet to HTML File, running the Applet, More About HTML Tags, Displaying Numerical Values, Getting Input from the User. [12 Hours]

Unit - V

Graphics Programming, Input/Output: Graphics programming: Introduction, The Graphics Class, Lines and rectangles, circles, and Ellipses, Drawing Arcs, Drawing Polygons, Lines Graphs, Using Control Loops in Applets, Drawing Bar Charts. Managing Input/Output Files in JAVA: Introduction, Concept of Streams, Stream Classes, Byte Stream Classes, Character Stream Classes, Using Streams, Other Useful I/O Classes, Using the File Class, Input / Output Exceptions, Creation of Files, Reading / Writing Characters, Reading / Writing Bytes, Handling Primitive Data Types, Concatenating and Buffering Files, Interactive Input and output, Other Stream Classes. [12 Hours]

Text Books:

1. A.Balaguruswamy, “Programming with JAVA”, A Primer, TMH, 1999.

Reference Books:

1. Thomas Boutel, “CGI programming in C and Perl”, Addison – Wesley, 1996.
2. Jefry Dwight et al, Using CGI, Second Edition, Prentice Hall, India, 1997.
3. Patrick Naughton & Herbert Schildt, JAVA 2: The Complete Reference, THM, 1999.
4. Schildt, “JAVA The Complete Reference”, 7th Edition.

BCA504T: OBJECT ORIENTED PROGRAMMING USING JAVA**BLUE PRINT****Question paper pattern for theory has two sections :**

Section – A :Contains 12 questions, out of which a student has to answer 10 questions.
Each question carries 2 marks ($10 \times 2 = 20$)

Section – B :Contains 5 full questions includes sub-question as (a) & (b). Each full question carries 10 marks ($5 \times 10 = 50$)

UNIT	CHAPTER	SECTION A 2 MARKS	SECTION B 10 MARKS
I	Introduction to JAVA	1	-
	Overview of JAVA Language	-	1
	Constants, Variables, and Data Types	1	1
	Operators and Expressions	1	1
	Decision Making and Branching	-	-
II	Classes, Objects and Methods	1	1
	Inheritance	-	1
	Arrays, Strings and Vectors	1	1
	Wrapper Class	-	-
III	Interfaces	1	1
	Packages	1	1
	Multithreaded Programming:	1	2
IV	Managing Exceptions	1	1
	Applet Programming	1	2

V	Graphics Programming	1	2
	Managing Input/Output Files in JAVA	1	1
	TOTAL	12	16
		ANSWER ANY 10	ANSWER ANY 5
	TOTAL MARKS	20	50

SECTION – A (2 Marks)

UNIT-I

[Nov / Dec 2015]

1. What do you mean by command line argument?
2. What are the two ways of giving values to the variable?
3. Write down the default values of byte and char data types?

[Nov / Dec 2016]

1. Why java is simple? Mention any two reasons?
2. What are string literals?
3. Give the general form of ‘switch’ statement?

[Nov / Dec 2017]

- 1.What is byte code? Justify how java is platform independent?
- 2.What is ‘labelled break’ and ‘labelled continue’?
- 3.Mention the data types in java?

[Nov / Dec 2018]

1. What are important elements of internet architecture?
2. What are the default values of float and char primitive data types in java?
3. Give the general form of switch statement?

[TMAQ – Important Tutor Mark Assignment Questions]

1. What is difference between JDK, JRE and JVM?
2. What are the different types of comments in Java?
3. What is the difference between Java and C++?
4. What is difference between Java and C?

5. What are the reserved literals in Java?
6. What is System.in and System.out?

SECTION – A (2 Marks)

UNIT-II

[Nov / Dec 2015]

1. Define a class and write down its syntax?
2. What is the use of ‘this’ and ‘super’ keyword?
3. How multiple inheritances are achieved in Java?

[Nov / Dec 2016]

1. What is instance variable? Give an example.
2. Write a few points about ‘default constructor’.
3. What does ‘static’ keyword do in a class?

[Nov / Dec 2017]

1. What is default constructor and parameterized constructor?
2. What is the use of ‘super’ and ‘this’ keyword?
3. Difference b/w ‘string’ class and ‘string buffer’ class.

[Nov / Dec 2018]

1. What is the difference between constructor and methods?
2. Difference between class and abstract class?

[TMAQ – Important Tutor Mark Assignment Questions]

1. What is an object? How to create an object?
2. How to get primitive values from wrapper objects?
3. What is explicit casting?
4. What is the difference between overloading and overriding.
5. What is a vector?
6. Why strings are called as immutable?
7. How is an array created in Java?
8. What is constructor? How constructor are overloaded?
9. What is inheritance? List types of Inheritance.

SECTION – A (2 Marks)

UNIT-III

[Nov / Dec 2015]

1. What is concurrency?

[Nov / Dec 2016]

1. What is java API?

[Nov / Dec 2017]

1. Define package. Mention its use.
2. Mention the ways of implementing multithreading in java.

[Nov / Dec 2018]

1. Mention any four thread methods?
2. What are the different access modifiers in java?

[TMAQ – Important Tutor Mark Assignment Questions]

1. What is Thread and Synchronization?
2. List all the access control modifiers in Java. Explain default access modifier.
3. What is an interface and abstract class?
4. What are the different ways of creating thread?

SECTION – A (2 Marks)

UNIT-IV

[Nov / Dec 2015]

1. What is exception?
2. How user defined exception is done?
3. Write down the applet code for “hello-class”file?
4. Why repaint () method is used?

[Nov / Dec 2016]

1. What is exception handling?
2. What is the need for ‘applet viewer’?
3. What is error? Compare with exception.
4. What is the purpose of ‘init ()’ method in applet?

[Nov / Dec 2017]

1. Define an exception. How is exception handling done in java.

[Nov / Dec 2018]

1. What is the difference between error and exception?
2. How applets differ from applications?

[TMAQ – Important Tutor Mark Assignment Questions]

1. How exceptions are classified in Java?
2. What is use of throw keyword?
3. Mention the attributes of PARAM tag?
4. Explain <APPLET> tag with an example.
5. Explain update() and repaint() methods.
6. How to create user defined exceptions?

SECTION – A (2 Marks)

UNIT-V

[Nov / Dec 2015]

1. Which method is used to draw a circle?

[Nov / Dec 2016]

- 1.What is the use of canvas in AWT?

[Nov / Dec 2017]

1. Mention any four classes in AWT package?
2. Define a stream in java. Briefly mention the broad classification of java stream classes?

[Nov / Dec 2017]

1. What is the use of java I/O classes?

[TMAQ – Important Tutor Mark Assignment Questions]

1. What is the difference between character oriented and byte oriented streams?
2. What is stream? How stream are classified?
3. What is the use of Graphic class?

SECTION – B (5 Marks)

UNIT-I

[Nov / Dec 2015]

1. Explain the features of java?
2. Write a note on scope of variables?
3. Explain the features of java?
4. Write a program to display all prime numbers between two limits using command line argument.

[Nov / Dec 2016]

1. Explain the line “public static void main(String args[])”.
2. Explain the history and evolution of java.
3. Explain bitwise and logical operators with example.

[Nov / Dec 2017]

1. What are static variables and static methods?

[Nov / Dec 2018]

1. Explain the difference between JDK and JRE.
2. Explain bitwise operators.

[TMAQ – Important Tutor Mark Assignment Questions]

1. What is the role of JVM and explain its components.
2. How to define constants in Java?
3. Explain operators in Java.
4. Explain Java tokens.
5. Explain the iterative statements in Java?
6. What is System.out.println()? Explain what is System.out.

SECTION – B(5Marks)

UNIT-II

[Nov / Dec 2015]

1. Differentiate between string and string buffer.
2. What is vector? Mention its advantages over an array.
3. How string class different from string buffer class? Give two methods of string class.
4. What is method overriding? Write a program to demonstrate method overriding.
5. Explain any seven string methods with example.
6. Write a note on inheritance.

[Nov / Dec 2016]

1. How to create object? What happens when you create objects?
2. Demonstrate ‘this’ keyword with simple java program.
3. Differentiate component and container class.
4. Give the general form of inheritance with one example.
5. Illustrate array declaration and accessing data elements using an example.
6. Differentiate constructors and methods.
7. Write a program to sort a list of elements in ascending order.

[Nov / Dec 2017]

1. Explain with example:
 - i. Method overloading.
 - ii. Method overriding
 - iii. Abstract method.
 - iv. Abstract class.
2. Define inheritance. Explain the types of inheritance supported by java.
3. Explain the tree string methods with example.
4. Differentiate between arrays and vectors.
5. Explain visibility control in java.

[Nov / Dec 2018]

1. What is the difference between overloading and overriding?
2. Explain any four string methods with example.

[TMAQ – Important Tutor Mark Assignment Questions]

1. Explain the various uses of wrapper classes.
2. Explain array of object references.
3. Why multiple inheritance is not supported in Java?
4. What are the rules of method overriding?
5. What is the use of abstract classes?
6. What is the use of finalize() method?
7. How to create wrapper object?

SECTION – B(5 Marks)

UNIT-III

[Nov / Dec 2015]

1. What is package? Write down the steps for creating user defined package.
2. What is thread ?explain thread cycle with neat diagram.
3. What is interface? Write a program to demonstrate interface.

[Nov / Dec 2016]

1. Give the general form of interface with one example.
2. Give the steps to create and use a java package with an example.
3. Explain the life cycle of thread.

[Nov / Dec 2017]

1. What is interface? Explain with an example how a class implements an interface.
2. Explain with an example the implementation of multithreading by extending ‘thread class’.

[Nov / Dec 2018]

1. Explain the process of creating user defined package with an example.
2. Write the steps involved in creating thread by implementing runnable interface.

[TMAQ – Important Tutor Mark Assignment Questions]

1. What is use of interface? What are the rules for defining an interface?
2. How multiple interface is achieved using interface?
3. What are access control modifiers?.
4. How to hide classes?
5. What are the different types of creating thread?
6. Explain thread priorities with an example.
7. Explain different method of thread class?

SECTION – B(5 Marks)

UNIT-IV

[Nov / Dec 2015]

1. What do u mean by unchecked exception? Write a program to illustrate try, catch and finally statement.
2. What is applet? Explain applet life cycle with a neat diagram.
3. Write a program to implement mouse events.

[Nov / Dec 2016]

1. Explain the steps to of executing an applet using a simple code.
2. Explain try, catch with example.

[Nov / Dec 2017]

1. Explain user defined exception in java.
2. Explain how parameters passed to an applet.

[Nov / Dec 2018]

1. Explain the life cycle of an applet with a neat diagram.

[TMAQ – Important Tutor Mark Assignment Questions]

1. How Java handles the exception? What are the types of exception in Java?
2. Name few of the exception classes in Java?
3. How to display numeric values in an applet?
4. Explain update() and repaint() methods.
5. What are the different ways of executing an applet?

SECTION – B(5 Marks)

UNIT-V

[Nov / Dec 2015]

1. Write down the steps for drawing polygons.
2. Give the classification of input stream classes
3. Write a note on graphics class and its methods.

[Nov / Dec 2016]

1. Write a short note on graphics class
2. Give classification on “java.io.IOException”. Explain IOException .

[Nov / Dec 2017]

1. Explain any seven methods of graphics class with an example for each.
2. Explain the use of file input stream class and file output stream class.

[Nov / Dec 2018]

1. Write a short note on data output stream and data output stream.

[TMAQ – Important Tutor Mark Assignment Questions]

1. Explain the graphics cocordinatesystem.?
2. How to draw bar chart in an applet?
3. What is BufferedInputStream and BufferedOutputStream?

V Semester B.C.A. Degree Examination, November/December 2015

(Y2K8 Scheme) (F+R)

COMPUTER SCIENCE

BCA – 504 : Java Programming

(70 – 2013 – 14 and Onwards) (60 – Prior to 2013 – 14)

Time : 3 Hours

Max. Marks : 60/70

Instructions : 1) Answer **all the Sections**.

2) **Section – D** is applicable to the students who were
admitted in 2013 – 14 and Onwards.

SECTION – A

Answer any ten questions :

(10x1=10)

1. What do you mean by command line argument ?
2. What are the two ways of giving values to a variable ?
3. Write down the default values of byte and char datatypes.
4. Define a class and write down its syntax.
5. What is the use of ‘this’ and ‘super’ keywords ?
6. How multiple inheritance is achieved in Java ?
7. What is concurrency ?
8. What is exception ?
9. How user defined exception is done ?
10. Write down the applet code for “hello-class” file.
11. Why repaint () method is used ?
12. Which method is used to draw a circle ?



SECTION – B

Answer **any five** questions : **(5×3=15)**

13. Explain Java program structure.
14. Write a note on scope of variables.
15. Differentiate between string and string Buffer.
16. What is a vector ? Mention its advantages over an array.
17. What is a package ? Write down the steps for creating user defined package.
18. How is a string class different from string buffer class ? Give two methods of string class.
19. Write down the steps for drawing polygons.
20. Give the classification of input stream classes.

SECTION – C

Answer **any five** questions : **(5×7=35)**

21. Explain the features of Java.
22. What is method overriding ? Write a program to demonstrate method overriding.
23. Explain any seven string methods with an example.
24. What is thread ? Explain thread life cycle with a neat diagram.
25. What is interface ? Write a program to demonstrate interface.
26. What do you mean by unchecked exception ? Write a program to illustrate try, catch and finally statements.
27. What is applet ? Explain applet life cycle with a neat diagram.
28. Write a note on graphic class and its methods.

SECTION – D

Answer **any one** question : **(1×10=10)**

29. a) Write a note on inheritance. 5
- b) Write a program to display all prime numbers between two limits using command line argument. 5
30. Write a program to implement mouse events.



V Semester B.C.A. Degree Examination, Nov./Dec. 2016
(CBCS) (2016-17 and Onwards)
COMPUTER SCIENCE
BCA 504 : Java Programming

Time : 3 Hours

Max. Marks : 70

Instruction: Answer all Sections.

SECTION – A

I. Answer any ten questions : (10x2=20)

- 1) Why Java is simple ? Mention any two reasons.
- 2) What are string literals ?
- 3) Give the general form of 'Switch' statement.
- 4) What is instance variable ? Give an example.
- 5) Write a few points about 'default constructor'.
- 6) What does 'static' keyword do in a class ?
- 7) What is Java API ?
- 8) What is Exception handling ?
- 9) What is the need for 'applet viewer' ?
- 10) What is error ? Compare with exception.
- 11) What is the purpose of 'init()' method in Applet ?
- 12) What is the use of canvas in AWT ?

SECTION – B

II. Answer any five questions : (5x10=50)

- | | |
|---|---|
| 13) a) Explain the line "public static void main (string args[])". | 4 |
| b) Explain the History and evolution of Java. | 6 |
| 14) a) How to create objects ? What happens when you create objects ? | 4 |
| b) Demonstrate 'this' keyword with Simple Java Program. | 6 |



- 15) a) Differentiate component and container class. 4
 b) Give the general form of interface with one example. 6
- 16) a) Explain bitwise and logical operators with examples. 4
 b) Illustrate array declaration and accessing data elements using an example. 6
- 17) a) Differentiate constructors and methods. 4
 b) Give the steps to create and use a Java package with an example. 6
- 18) a) Explain the life cycle of a Thread. 4
 b) Explain the steps of executing an Applet using a simple code. 6
- 19) a) Explain try catch with an example. 4
 b) Write a short note on Graphics Class. 6
- 20) a) Write a program to sort a list of elements in ascending order. 4
 b) Give the classification on "java.io.IOException". Explain IOException. 6

SECTION - B

(02=0x0)

- A. Explain the following terms:
 a) Inheritance
 b) Polymorphism
 c) Encapsulation
 d) Abstraction
 e) Java Programming

.O.T.P

V Semester B.C.A. Degree Examination, November/December 2017
(F+R) (CBCS)
(2016 – 17 & Onwards)
BCA 504 : JAVA PROGRAMMING

Time : 3 Hours

Max. Marks : 70

Instruction : Answer all Sections.**SECTION – A**

I. Answer **any ten** questions : **(10×2=20)**

- 1) What is bytecode ? Justify how Java is platform independent.
- 2) What is default constructor and parameterized constructor ?
- 3) What is 'labelled break' and 'labelled continue' ?
- 4) Define a package. Mention its use.
- 5) Mention the ways of implementing multithreading in Java.
- 6) Mention any four thread methods.
- 7) Define an exception. How is exception handling done in Java.
- 8) Mention any four classes in AWT package.
- 9) Define a stream in Java. Briefly mention the broad classification of Java stream classes.
- 10) What is the use of 'super' and 'this' keyword ?
- 11) Mention the datatypes in Java.
- 12) Differentiate between 'string' class and 'string buffer' class.

SECTION – B

II. Answer **any five** questions : **(5×10=50)**

- | | |
|---|---|
| 13) a) Explain the features of Java. | 7 |
| b) What are static variables and static methods ? | 3 |



- 14) Explain with an example :
i) Method overloading. 3
ii) Method overriding. 3
iii) Abstract method. 2
iv) Abstract class. 2
- 15) a) Define inheritance. Explain the types of inheritances supported by Java. 7
b) Explain any three string methods with examples. 3
- 16) a) Differentiate between arrays and vectors. 3
b) Explain visibility control in Java. 7
- 17) a) What is an interface ? Explain with an example how a class implements an interface. 6
b) Explain user defined exceptions in Java. 4
- 18) a) Explain the cycle of a thread with a neat diagram. 6
b) Explain with an example the implementation of multithreading by extending 'Thread' class. 4
- 19) a) Explain life cycle of an applet with a neat diagram. 7
b) Explain how parameters are passed to an applet. 3
- 20) a) Explain any seven methods of graphics class with an example for each. 7
b) Explain the use of FileInputStream class and FileOutputStream class. 3

V Semester B.C.A. Degree Examination, November/December 2018**(F+R) (CBCS)****(2016 – 17 & Onwards)****Computer Science****BCA 504 : JAVA PROGRAMMING**

Time : 3 Hours

Max. Marks : 70

Instruction : Answer all Sections.**SECTION – A****I. Answer any ten questions : (10×2=20)**

- 1) What are important elements of internet architecture ?
- 2) What are the default values of float and char primitives data types in Java ?
- 3) Give the general form of switch statement.
- 4) What is the difference between constructor and method ?
- 5) What is the difference between class and abstract class ?
- 6) What is instance variable ? Give an example.
- 7) Mention any four thread methods.
- 8) What are the different access modifiers in Java ?
- 9) What is the difference between error and exception ?
- 10) How applets differ from applications ?
- 11) What is the use of Java I/O classes ?
- 12) Define a stream in Java. Briefly mention the broad classification of Java stream classes.

SECTION – B**II. Answer any five questions : (5×10=50)**

- 13) a) Explain the features of Java. 7
- b) Explain the difference between JDK and JRE. 3

P.T.O.



- 14) a) Explain bitwise operators. 5
b) What is the difference between overloading and overriding ? 5
- 15) a) Explain any four string methods with examples. 4
b) Define inheritance. Explain any two types of inheritance supported by Java with examples. 6
- 16) a) Explain the process of creating user defined package with an example. 6
b) Give the general form of interface with an example. 4
- 17) a) Explain the cycle of a thread with neat diagram. 5
b) Write the steps involved in creating thread by implementing runnable interface. 5
- 18) a) Explain try...catch with an example. 4
b) Explain life cycle of an applet with a neat diagram. 6
- 19) a) Explain the steps of executing an applet using a simple code. 5
b) Write a program to implement mouse events. 5
- 20) a) Explain any six methods of graphics class with an example for each. 6
b) Write a short note on data output stream and data input stream. 4

SECTION - B

(Ques-01)

T

2

0.74

BCA-504 JAVA PROGRAMMING (2 marks)

1. What are the two ways of giving values to the variable?

Ans. The two ways of giving values to the variable are:

- By using an assignment statement
- By using a read statement

2. Write down the default values of byte and char data types?

- The default value of byte=0
- The default value of char=NULL CHARACTER

3. What do you mean by command line argument?

Ans. Command line arguments are the parameters that are supplied to the application program at the time of invoking it for execution.

4. Define a class and write down its syntax?

Ans. A class is a template that defines the form of an object. A class definition consists of two members- Data member and Methods.

Syntax:

```
[access modifier] [class modifier] class class_name [extends super_class_name]  
[implementinterface list]  
{  
[variable declaration]  
[method declaration]  
}
```

5. What is the use of ‘this’ and ‘super’ keyword?

Ans. Use of ‘this’ keyword:

- This refers to the current object.
- The first use of this is to call constructor from another constructor, specially one in the current class. The second function is to avoid namespace conflicts between a methods or constructors parameter list and its variable.

Use of ‘super’ keyword:

- The super is used to call super class constructor explicitly.
- If the same variables are defined in both super class and subclass, then super is used to access super class variable inside subclass method.
- If the same methods are present in both super class and subclass, then super is used to call super class method from subclass method.

6. How multiple inheritances are achieved in Java?

Ans. Multiple inheritance in Java programming is achieved or implemented using interfaces. Java does not support multiple inheritances using classes. A class can extend only one class but it can implement multiple interfaces.

7. What is concurrency?

Ans. Concurrency is the ability to run several programs or several parts of a program in parallel. The backbone of java concurrency is threads. A thread is a lightweight process which has its own call stack, but can access shared data of other threads in the same process.

8. What is exception?

Ans. An exception is an error that occurs at run time.

9. How user defined exception is done?

Ans. To create the exception object, the program uses the throw keyword followed by the instantiation of the exception object. At runtime, the throw clause will terminate execution of the method and pass the exception to the calling method.

10. Write down the applet code for “hello-class”file?

Ans.

```
import java.applet.Applet;  
import java.awt.Graphics;  
  
public class HelloCLASSApplet extends Applet  
{  
    public void paint(Graphics g){  
        g.drawString("Hello CLASS", 50, 50);  
    }  
}
```

11. Why repaint () method is used?

Ans. This method can't be overridden. It controls the updated() -> paint() cycle. You should call this method to get a component to repaint itself. If you have done anything to change the look of the component, but not its size (like changing color, animating, etc.) then call this method.

12. Which method is used to draw a circle?

Ans. drawOval() is the method used to draw a circle.

13. Mention the ways of implementing multithreading in Java?

Ans. The ways of implementing multithreading in Java are:

- Extends Thread class. Create a thread by a new class that extends Thread class and create an instance of that class.
- Implementing the runnable interface. The easiest way to create a thread is to create a class that implements the runnable interface.

14. Mention any four thread methods?

Ans. Thread methods are:

- **start()**
- **run()**
- **yield()**
- **wait()**

15. Mention any four classes in AWT package?

Ans.

- **GUI Component classes**
- **GUI Container classes**
- **Layout managers**
- **Custom graphics classes**

16. Why java is simple? Mention any two reasons.

Ans. Java is one of the simple programming language because java removes some complex concept like pointer and execution time will be less.

Reasons:-

- Much easier to write bug free code.
- Java has considerably more functionality than c.

17. What are string literals?

Ans. String literals are a sequence of characters from the source character set enclosed in double quotation marks (" ").

18. Give the general form of ‘switch’ statement.

Ans.

switch(expression)

{

```
    Case value1:  
        break;  
    Case value2:  
        break;  
    .....  
    default:  
}
```

19. What is instance variable? Give an example.

Ans.

- Instance variables are declared in a class , but outside a method. They are also called member or field variables.
- Instance variables are created when an object is created and destroyed when the object is destroyed.

20. Write a few points about ‘default constructor’.

Ans. Constructor with no arguments is called as default constructor

The default constructor is useful to initialize all objects with same data.

Ex:

A0

{

}

21. What does ‘static’ keyword do in a class?

Ans. The keyword static indicates that the particular member belongs to a type itself, rather than to an instance of that type. This means that only one instance of that static member is created which is shared across all instances of the class.

22. What is java API?

Ans. Java API(application programming interface) is the set of huge number of classes and methods grouped into packages and it is included with the java development environment. The most commonly used packages are:

- java.lang
- java.io

23. What is exception handling?

Ans. The java mechanism that deals with handling the errors in an organized fashion is called as the exception handling.

24. What is the need for ‘applet viewer’?

Ans. An applet viewer is tool provided with the standard java JDK to execute an applet.

25. What is error? Compare with exception.

Ans. Errors are related to errors that occur in the java virtual machine itself, and not in a program. These types of exceptions are beyond our control, and a program will not handle them. An exception is an error which can be handled .An error is an error which cannot be handled.

26. What is the purpose of ‘init ()’ method in applet?

Ans. init()- is called to initialize the applet before it gets loaded.

27. What is the use of canvas in AWT?

Ans. A canvas is for drawing on, basically. It also serves like a Panel for creating a custom AWT-based component, but unlike Panel it can't contain other components.

28. What is byte code? Justify how java is platform independent.

Ans. Byte code are the machine level language of the JVM. when a JVM loads a class file, it gets one stream of bytecodes for each method in the class. Java is a platform independent because java compiler converts the source code to byte code. It can be executed on any platform using JVM.

29. What is default constructor and parameterized constructor?

Ans. Default constructor: Constructor with no arguments is called as default constructor

Ex:

```
A()  
{  
}
```

Parameterized constructor: Constructor with arguments is called as parameterized constructor.

Ex:

```
A(int a)  
{  
}
```

30. What are the default values of float and char primitive data types in java?

Ans.

- Default value of float =0.0f
- Default value of char = NULL CHARACTER

31. What is ‘labelled break’ and ‘labelled continue’?

Ans. Labeled break: the break statement breaks out the closest loop or switch statement.

Ex:

```
for(int i=0;i<10;i++){  
    while(true){  
        break;  
    }  
}
```

Labelled continue: the continue statement transfers the control to the closest enclosing loop.

Ex: *for(int i=0;i<10;i++){*

while(j<10){

if(j==5)

continue;

}

}

32. Define package. Mention its use.

Ans. Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. Packages are used for preventing naming conflicts.

33. What are important elements of internet architecture?

Ans.

- Modem
- ISP
- Router
- Internet
- Backbone
- TCP/IP
- Network access points or NAPs
- Domain name service(DNS) and DNS Servers

34. Give the general form of switch statement.

Ans.

```
switch (expression)
{
    Case value1:
        Break;
    Case value2:
        Break;
    .....
    default:
}
```

35. What is the difference between constructor and methods?

Constructor	Methods
Constructor is used to initialise the instance variables of a class	A Method is used for any general purpose tasks like calculations
A constructor name should be always same as class name	The method and class name can be same or different

36. Difference between class and abstract class?

Class	Abstract class
The class does not contain abstract methods	It contains abstract methods
The class can be instantiated	Abstract classes cannot be instantiated

37. What is instance variable? Give an example?

Ans. Instance variable are the variable declared in a class but outside a method

Ex: *class rectangle*

```

{
    double length;
    double breadth;
}

```

38. Mention any four thread methods?

Ans.

- CurrentThread()
- getName()
- run()
- sleep()

39. What are the different access modifiers in java?

Ans.

- Private
- Default
- Protected
- Public

40. What is the use of java I/O classes?

Ans. Java I/O (Input and Output) is used to process the input and produce the output. Java uses the concept of a stream to make I/O operation fast.

41. What is the difference between error and exception?

ERROR	EXCEPTION
All errors in java are unchecked type	Exception s include both checked as well as unchecked type
Errors are mostly caused by the environment In which applications is running	exception are mainly caused by application itself

42. Define a stream in java. Briefly mention the broad classification of java stream classes?

Ans. A stream can be defined as a sequence of data. The input stream is used to read data from a source and the Output Stream is used for writing data to a destination.

43. How applets differ from applications?

Ans. The main difference between Applet and Application is that the applet is a small java program that can be executed by a Java-compatible web browser while the application is a standalone program that can directly run on the machine.

44. What is the use of ‘super’ and ‘this’ keyword?

Ans. super : it is used to call the constructors of super class.

this : it is used to call the constructors of same class.

45. Mention the data types in java?

Ans.

Primitive data types

Non-Primitive data types

46. Difference b/w ‘string’ class and ‘string buffer’ class.

STRING CLASS	STRING BUFFER
It is immutable	It is mutable
It is slow and consumes more memory	It is fast and consumes less memory

5 MARKS QUESTIONS

1. Explain the features of java?

Ans: Features of java are:

- Simple: Easy to learn. Because java inherits the c/c++ syntax and many of the object oriented features of c++.
- Secure: Java provides a firewall between a networked application and user computer.
- Portable: Translating a java program into bytecode makes it much easier to run a program in a wide variety of environments.
- Object oriented: Java enhances and refines the object oriented paradigm used by C++. The object model in java is simple and easy to extend.
- Robust: Java eliminates memory management problems by managing memory allocation and deallocation.
- Multithreaded: Java supports multithreaded programming which allows you to write programs that do many things simultaneously.
- Dynamic: Java programs have run time information that is used to verify and resolve accesses to objects at run time.
- Distributed: Java is designed for the distributed environment of the internet, because it handles TCP/IP protocols.
- Interpreted and Performance: Java bytecode can be interpreted on any system that provides a high JVM. Java is well designed to perform well on very low power CPUs.
- Architecture Neutral: Java programs can be executed on any processors like Pentium, Celeron, dual core, AMD and so on,hence it is called architecture neutral.

2. What are static variables and static methods?

Ans: Static variables:

- The instance variables are non-static and it is part of an object. But static variables are special type of variables that are not associated with an object, they are associated with class.
- The static variables are also called as class variables.
- The static variables can be accessed without an object.
- A static variable can be accessed directly by the class name and does not need any object.

Syntax: <class_name>.<variable_name>

Example:

```
class staticdemo{  
    int x,y;  
    static int z;
```

```
System.out.println(staticdemo.z);  
}
```

Static Methods:

- The methods can also be declared as static. A static method is associated with a class rather than the instances.
- The static methods are also called as class members.
- The most common example of a static member is main(). The main() is declared as static because it must be called by the operating system when our program begins.
- A static methods can be accessed directly by the class name and does not need any object.

Syntax:

<class_name>.<method_name>(arguments)

Example:

```
staticdemo.method1();
```

Declaration:

```
class staticdemo{  
int x,y;  
static int z;  
}
```

3. Explain any three string methods with example?

- concat(): This method creates a new string by appending the contents of string object passed as arguments to the contents of string on which the method is invoked.

Example:

```
public String concat(String str)  
String str="Skyward";  
System.out.println(str.concat("Publishers")); // "Skyward Publishers" is printed.
```

- replace(): This method creates a new string using the same contents as that of the string object on which the method is invoked.

Example:

```
public String replace(char old,char new)  
String original="Java ProgrAmming";
```

`System.out.println(original.replace('a','o'));` //”Java Programming” is printed.

- **substring():** The substring method creates a new string using partial contents of the string on which it is invoked. This method has two overloaded version.

Example:

```
public String substring(int begin)
public String substring(int begin,int end)
String original="watermelon";
System.out.println(original.substring(5)); // prints “rings”
```

4. Differentiate between arrays and vectors?

Ans:

Sr. No.	Vector	Array
1	Vector can grow and shrink dynamically.	Array can't grow and shrink dynamically.
2	Vector can hold dynamic list of objects or primitive data types.	Array is a static list of primitive data types.
3	Vector class is found in <code>java.util</code> package .	Array class is found in <code>java.lang</code> (default) package.
4	Vector can store elements of different data types.	Array can store elements of same data type.
5	Vector class provides different methods for accessing and managing vector elements.	For accessing element of an array no special methods are available as it is not a class, but a derived type.
6	Syntax : <code>Vector objectname=new Vector();</code>	Syntax : <code>datatype arrayname[]=new datatype[size];</code>
7	Example: <code>Vector v1=new Vector();</code>	Example: <code>int[] myArray={22,12,9,44};</code>

5. Explain visibility control in Java?

Ans: Visibility controls in Java are:

Private: The private modifiers specifies the most restrictive access level, it can apply it to methods and member variables. The private methods and variables are accessible only within the declaring class.

- A top level class cannot be declared as private, as it would mean that the class cannot be instantiated and it would obviously unreasonable to declare a class that can never be accessible.

Example:

```
class A{
private int data=40;
private void msg(){System.out.println("Hello java");}
}
```

```

public class Simple{
public static void main(String args[]){
    A obj=new A();
    System.out.println(obj.data);//Compile Time Error
    Obj.msg();//Compile Time Error
}

```

Default: When a class,method or variable declaration does not have any access modifier, it will have a default or package level. A class or its members with default access are available to all the classes within the same package in which the class is declared.

EXAMPLE:

```

package pack;
class A{
    void msg(){System.out.println("Hello");}
}

//save by B.java

package mypack;
import pack.*;
class B{
    public static void main(String args[]){
        A obj = new A();//Compile Time Error
        Obj.msg();//Compile Time Error
    }
}

```

Protected: The protected modifiers is applicable only to methods and member variables. A top level class cannot be declared as protected.The member variables and methods of a class,declared as protected are accessible to all classes in same package and all the subclasses of the declaring class in other package.

EXAMPLE:

```

package pack;
public class A{
    protected void msg(){System.out.println("Hello");}
}
package mypack;
import pack.*;
class B extends A{

```

```

public static void main(String args[]){
    B obj = new B();
    Obj.msg();
}
}

```

Public: The public modifiers is the least restrictive of all access modifiers. It can apply it to a class, its methods and its member variables. A public class can be instantiated without any restrictions.

EXAMPLE:

```

package pack;
public class A{
    public void msg(){System.out.println("Hello");}
}
//save by B.java

```

```

package mypack;
import pack.*;

class B{
    public static void main(String args[]){
        A obj = new A();
        Obj.msg();
    }
}

```

6. What is Interface? Explain with an example how a class implements an interface.

Ans: An interface is a description of a set of abstract methods that is supported to be implemented by the classes. In an interface no method can include a body. It specifies what can be done, but no implementation. Once an interface is defined any number of classes can be implemented. The interface can be defined using the interface keyword.

Syntax:

```

interface interface_name {
    public static variables
    public abstract methods
}

```

Example:

```
interface XYZ
{
    public void functionx();
    public void functiony();
}

class ABC implements XYZ
{
    public void functionx() {}
    public void functiony() {}
}
```

7.Explain user defined exception in Java?

Ans: Java provides us facility to create our own exceptions which are basically derived classes of exception. Things to remember before writing an exception

- All exceptions must be a child of Throwable
- To write a runtime exception, extend the RuntimeException class.

Steps to define our own exception.

- Extend the exception class
- Setup the constructor. The constructor takes the variable that is used to tell the user that the number is incorrect.
- Create the function that returns the error to the user and override `toString()` method.
- When the error occurs, the code throws an exception.

Example:

```
class MyException extends Exception{
    String str1;
    MyException(String str2){
        str1=str2;
    }
    public String toString(){
        return("MyException Occurred:"+str1);
    }
}
```

```

}

class Example1{
    public static void main(String args[]) {
        try{
            System.out.println("Starting of try block");
            throw new MyException("This is my error message");
        }
        catch(MyException exp) {
            System.out.println("Catch block");
            System.out.println(exp);
        }
    }
}

```

8.Explain with an example the implementation of multithreading by extending Thread class.

Ans: The implementation of multithreading by extending three Thread classes ThreadA, ThreadB, ThreadC. Then create three thread objects on ThreadA, ThreadB, ThreadC. Attach these three object to t1, t2, t3. The following example is printing the numbers 1 to 20. ThreadA prints the value from 1 to 5, threadB prints the value from 6 to 10, thread prints the value from 11 to 15 and the main class prints the value from 16 to 20.

Example:

```

class ThreadA extends Thread {
    public void run() {
        for(int i=1;i<=5;i++)
            System.out.println(Thread.currentThread().getName()+"="+i);
        System.out.println("End of Thread One");
    }
}

class ThreadB extends Thread {
    public void run() {
        for(int i=6;i<=10;i++)

```

```

System.out.println(Thread.currentThread().getName()+"="+i);
System.out.println("End of Thread Two");
}

}

class ThreadC extends Thread {
public void run() {
for(int i=11;i<=15;i++)
System.out.println(Thread.currentThread().getName()+"="+i);
System.out.println("End of Thread Three");
}
}

class MultipleThreads {
public static void main(String args[]) {
ThreadA ta = new ThreadA();
ThreadB tb = new ThreadB();
ThreadC tc = new ThreadC();
Thread t1=new Thread(ta,"Thread One");
Thread t2=new Thread(tb,"Thread Two");
Thread t3=new Thread(tc,"Thread Three");
t1.start();
t2.start();
t3.start();
for(int i=16;i<=20;i++)
System.out.println(Thread.currentThread().getName()+"="+i);
System.out.println("End of Main");
}
}

```

9.Explain the cycle of thread with a neat diagram?

Ans: Start(): When the start() method is called, the thread enters in a ready- to- run state. This thread is now in the pool of threads ready for the execution. Sometimes

later the thread is picked up by the scheduler for execution and moved to the running state.

- The thread scheduler may move the thread out of the running state even if it has not finished the execution.

Ready-to-run state(Runnable): A thread is in ready-to-run state when it is eligible to run, but the scheduler has not yet picked it to actually run. A thread can also return to ready-to-run state either after running or after the not-ready-to-run state.

- Call the start() method only once on a thread object. Re invoking a start method on thread object that is already started is illegal.

Running: When the thread enters in this state, the JVM starts to execute the thread run() method. The thread remains in this state and keep running until it is either swapped out by thread scheduler or it voluntarily give up its turn for some reasons.

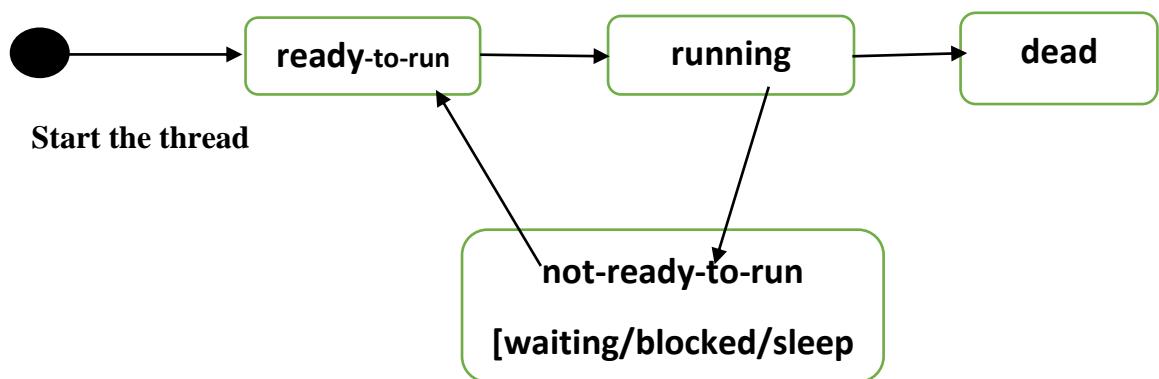
Not-ready-to-run (Blocked state): A thread moves out of the running state when it is waiting for something to happen.

Sleeping: We may want a thread to do-nothing for some time. We can call Thread.sleep() method in the thread run() method. This method tells the currently running thread to sleep for some period of time.

Waiting: Sometimes a thread might wait(), just because we have asked it to wait in its run method. In that case, the thread changes its state from running to waiting .

Blocked: Sometimes a thread needs to wait for a resource while running. For instance ,if it is reading from a network resources in its run method, it has to wait until that resource becomes available.

Dead state: A java thread enters this state when it has finished the execution of its run method. We cannot start the thread once it is dead. The thread can be started only once in its life time. If we re-invoke start() on a thread which is dead,it does not start again. Using isAlive() method we can test whether the thread is alive or dead.



10.Explain how parameters are passed to an applet?

Ans: Applet can get different input from the HTML file that contains the <APPLET>tag through the use of applet parameter. To set up and handle parameters in the applet , we need two things:

1. A special parameter tag in the HTML file.
2. Code in our applet to read those parameters.

Example:

```
import java.applet.Applet;  
import java.awt.Font;  
import java.awt.Graphics;  
  
public class MyFontApplet extends Applet {  
  
String fontName;  
int fontSize;  
  
public void init() {  
fontName = getParameter("font");  
fontSize = Integer.parseInt(getParameter("size"));  
}  
  
Public void paint (Graphics g) {  
Font f = new Font(fontName and fontSize);  
g.setFont(f);  
g.drawString("Skyward Publishers",50,50);  
}  
}
```

11.Explain any seven methods of graphics class with an example for each?

Ans:

METHODS	DESCRIPTION
draw3Drect()	Draws a 3-D rectangle
drawArc()	Draws an arc
drawLine()	Draws a line
drawOval()	Draws an oval
fillArc()	Draws a filled arc
fillOval()	Draws a filled oval
fillRect	Draws a filled rectangle

- **Example for drawLine()**
`import java.awt.Graphics;
public class DrawLineDemo extends java.applet.Applet {
 public void paint(Graphics g) {
 g.drawLine(25,25,75,75);
 }
}`
- **Example for draw3DRect()**
`import java.awt.Graphics;
public class DrawLineDemo extends java.applet.Applet {
 public void paint(Graphics g) {
 g.draw3DRect(20,20,60,60,true);
 g.draw3DRect(120,20,60,60,false);
 }
}`
- **Example for drawArc() and fillArc()**
`import java.awt.Graphics;
public class DrawLineDemo extends java.applet.Applet {
 public void paint(Graphics g) {
 g.drawArc (50,50,80,60,45,120);
 g.fillArc(150,50,80,60,45,120);
 }
}`

12.Explain the use of FileInputStream class and FileOutputStream class?

Ans: Use of FileInputStream:

- A file is opened for input by creating a **FileInputStream** object.

FileInputStream(String fileName) throws **FileNotFoundException**

Here, filename specifies the name of the file you want to open. If the file does not exist, then **FileNotFoundException** is thrown.

Example:

```
FileInputStream fis;  

try {  

    fis = new FileInputStream("myFile.dat");  

} catch(IOException) {  

}  


```

Use of FileOutputStream:

- To open a file for output, create a **FileOutputStream** object.

FileOutputStream(File file) throws **FileNotFoundException**

Here, creates a file output stream to write to the file represented by the specified file object. If the file cannot be created, then `FileNotFoundException` is thrown.

Example:

```
File f = new File("myFile.dat");
FileOutputStream fos;
try {
    fos = new FileOutputStream(f);
} catch(IOException) {
}
```

13. Explain with example:

Method overloading: Method Overloading means to have two or more methods with same name in the same class with different arguments.

Example:

```
class Myclass {
    public void getAmount (int rate) {...}
    public void getAmount (int rate, long principal) {...}
}
```

Method overriding: Method overriding occurs when sub class declares a method that has the same type arguments as a method declared by one of its super class.

Example:

```
class Baseclass {
    public void getAmount (int rate) {...}
}

class Myclass extends Baseclass{
    public void getAmount (int rate) {...}
}
```

Abstract method: A method without body is known as abstract method. A method must always be declared in an abstract class.

Example:

```
abstract void method1(); // Abstract method
```

Abstract class: Abstract classes are classes that contain one or more abstract methods.

Example:

```
abstract class Test {           // Abstract class
    int a,b,c;
    abstract void method1();     // Abstract method
    abstract void method2();     // Abstract method
    void method3() {
    }
}
```

14. Write a short notes on data output stream and data input stream?

Input stream

- Input stream is an abstract class that provides the framework from which all the other input streams are derived
- We cannot create an instance of `InputStream` class as it is abstract class .
- Whenever we want to read the data in bytes format, then we use the input stream classes
- The `InputStream` class contains lot of methods for reading bytes , closing stream , skipping part of data in the streams. finding the number of bytes present in the input data ,etc.

Output stream

- Output stream is an abstract class that provides the framework from which all other output streams are derived
- We cannot create an instance of `OutputStream` class as it is abstract class
- Whenever we want to write data in byte format, then we use the output stream classes.
- The `OutputStream` class contains lot of methods for writing bytes, closing streams ,etc.

15. Write a program to implement mouse events.

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class AppletMouseListener extends Applet implements MouseListener
{
    String str="";
}

public void init()
```

```

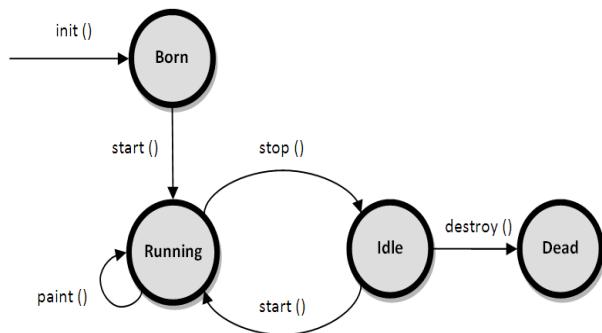
{
public void mousePressed(MouseEvent e)
{
str = "You pressed mouse";
repaint();
}
public void mouseReleased(MouseEvent e)
{
str = "You released mouse";
repaint();
}
public void mouseClicked(MouseEvent e)
{
str = "You clicked mouse";
repaint();
}
public void mouseEntered(MouseEvent e)
{
str = "Mouse entered frame";
repaint();
}
public void mouseExited(MouseEvent e)
{
str = "Mouse existed frame";
repaint();
}
public void paint(Graphics g)
{
g.drawString(str, 75, 150);
}
}

```

16. Explain the life cycle of an applet with a neat diagram?

- An applet is born with init() method and starts executing with start() method. To stop the applet, stop() method is called and to terminate the applet destroy() method is called. Once the applet is terminated we should reload the HTML page to get the applet start once again from init() method this way of executing the methods are called as life cycle of an applet.

Applet life cycle diagram



- **init()**: The first method called by an applet once it has been loaded by the browser. It is called before the applet begins execution, and can be overridden to perform any initialization tasks. When the method execution is completed, browser looks for the next method start().

Syntax:

```
public void init()
{
-----
}
```

- **start()**: start() is automatically called to begin the execution of the applet. This method is called each time the applet is revisited by the user.

Syntax:

```
public void start()
{
-----
}
```

- **stop()**: This method is called by the browser when an applet is stopped.

Syntax:

```
public void stop()
{
-----
}
```

- **destroy()**: this method is called when an applet is being terminated from the memory. The stop() method will always be called before destroy(). The code related to releasing memory allocated to the applet should be done in this method.

Syntax:

```
public void destroy()
{
-----
}
```

17. Difference between JDK and JRE?

JDK	JRE
It is called java development tool kit environment	It is called java runtime
JDK is needed for developing Java application	JRE is a plug in needed for running java programs
JDK can be downloaded, supported freely from java.sun.com	JRE can be downloaded, supported freely from Java.com
JDK needs more disk space as it contains JRE along with various	JRE is smaller than JDK so it needs less disk

Development tools	space
It is bundle of software that you can be used to develop java Based application	It is an implementation of java virtual machine which actually executes java Program

18. Explain bitwise operator?

Ans Bitwise operator works on bits and performs bit by bit operations. The operations on the bits are performed on 1s and 0s only. This means that any number is decimal or hexadecimal format involved in a bit operation must be converted to binary first

OPERATOR **DISCRIPTION**

& binary AND operator copies a bit to the result if it

Exists in both operands

| binary OR operator copies a bit if it exists in either

Operands

^ binary XOR operator copies the bit if it is set in one

Operand but not both

~ binary 1s complement operator is unary and has the

Effect of negating bits

<<binary left shift operator . this left operands value is

Moved left by the number of bits specified by the right

Operand

>>binary right shift operator . this left operands value is

Moved right by the number of bits specified by the

Right Operand .

19. What is the difference b/w overloading and overriding?

Overloading

1. Signature has to be different just a difference in return type is not enough.
2. Any access modifier can be used.
3. The methods exception list may vary freely
4. The method to be called will be decided at the time of compilation.
5. Methods can be static or non-static.

Over Ridding

1. Signature can be same.
2. Over ridding method cannot be more restrictive than the over ridden method.
3. Over ridding method may not throw more checked exceptions than the overridden method.
4. The method to be called will be decided at the time of run time based on type of the object.
5. Static method don't participate in over ridding since they are resolved at compile time based on the type of reference variable.

20. What is inheritance explain two types of inheritance?

- Inheritance can be defined as the process of acquiring properties of one object from other object.
 - Inheritance is one of the important principles of object oriented programming because it allows creating a new class based on the class that has already been defined using inheritance.
 - We can create a general class that defines common functionality this class can be inherited by another classes each class can add new functionality that are unique to it.
 - Types of inheritance
- a. Multilevel Inheritance: A class extending another class as in hierarchical structure is termed as multilevel inheritance.

Eg: `class A {
 int x,y;
 method1()
}
class B extends A{
 int i,j;
 method2()
}
class C extends B {
 int a,b;
 method3()
}`

- b. Single Inheritance : A class extends from another class.

Eg: `class A {
 int x,y;
 method1()
}
class B extends A{
 int i,j;
 method2()
}`

- c. **Hierarchical Inheritance:** Many class extending from single super class. One super class and many sub classes.

Eg:

```
class A {  
    int x,y;  
    method1( )  
}  
class B extends A{  
    int i,j;  
    method2( )  
}  
class C extends A {  
    int a,b;  
    method3( )  
}
```

21. Explain the line “public static void main(string args[])”

- **main() method:** main() method is the starting point for JVM to start execution of a java program.
- **public:** public keyword is an access specifier in this case, main() must be declared as public, since it must be called outside of its class when the program is started.
- **static:** static method can be called and executed without creating an object of class, since we want to call main() method without creating any object, we should declare main() as static.
- **void:** The keyword void simply tells the compiler that main() does not return a value.
- **String args[]:** String is java's predefined class and args[] is the name of an array.

22. Explain the history and evolution of java.

Ans. Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time. The history of java starts with Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was suited for internet programming. Later, Java technology was incorporated by Netscape.

23) How to create objects? What happens when you create objects?

An object is created by instantiating a class. The process of creating an object of a class is called as instantiation and created object is called as an instance.

- To create a new object, java uses the new keyword
- The object are created using the new operator with the name of the class we want to create an instance of, then parentheses after that. The general form of creating an object is

<classname><reference-variable>=new<classname>([arguments])

- When an object is created an instance of a class is created. Reference variable does not define an object but it is simply a variable that can refer to an object. The new operator dynamically allocates memory for an object and returns a reference to it.
- Eg: Account acc = new Account();

Where Account is a class name, acc is reference variable and new is the operator to create an object. The Account object is created in the heap memory. The address of that object is assigned to the reference variable acc. The reference variable is declared in the stack.

24) Demonstrate 'this' keyword with simple java program.

- This keyword is used to refer the current object
- It is used to call the constructors of same class

```
class Test
{
    int a;
    int b;

    Test(int a, int b)
    {
        this.a = a;
        this.b = b;
    }

    void display()
    {
        System.out.println("a = " + a + "b = " + b +);
    }

    public static void main(String[] args)
    {
        Test object = new Test(10, 20);
        object.display();
    }
}
```

25. Differentiate component and container class.

Component class:

- The component class is found under java. AWT package.
- The container class is the subclass of component class.
- All non-menu related elements that comprise a graphical user interface are derived from the abstract class component.
- The component class defines a number of methods for handling events, changing window bounds, controlling fonts and colours, and drawing components, and their content.

Container class

- A container is a components that that can accommodate other components and also other containers.
- Containers provides the support for building complex hierarchical graphical user interface
- Container provides the overloaded methods add() to include components in the container

26) Explain logical operators with example.

Logical operators return a true or false value based on the state of the variables.

The logical operators are

operator	Description	Example(A=true and B=false)
& (AND operator)	If both the operands are non zero then the condition becomes true.	(A&B) is false
&& (Short circuit AND operator)	If both the operands are non zero then the condition becomes true.	(A&&B) is false
(OR operator)	If any of the two operands are non zero then the condition becomes true	(A B) is true
(Short circuit OR operator)	If any of the two operands are non zero then the condition becomes true.	(A B) is true
^ (XOR operator)	This return true only if its operand. If its operand are different otherwise false.	(A^B) is true
! (NOT operator)	Use to reverses the logical state of its operand. If a condition is true then logical NOT operator make false.	!(A&&B) is true

27. Illustrate array declaration and accessing data elements using an example.

Array is collection of elements of same type. The array stores a fixed-size sequential collection of elements of the same type.

Declaring an array

An array is declared by specifying the datatype of elements it is going to hold.

The array declaration is usually the data type followed by a pair of square brackets followed by the name of the array

General form of declaring an array

```
Datatype[ ] arrayName;
```

Example :

```
int[ ] numbers;  
accessing data elements using array  
ex:    create an integer array and display the elements of an array  
public class simpleArray  
{  
    Public static void main(String args[])  
    {  
        Int[ ] numbers = {10,20,30,40,50};  
        for (int a : numbers)  
        {  
            System.out.println(a);  
        }  
    }  
}
```

28. Differentiate constructor and methods

Constructor

- The constructor is used to initialize the instance variables of a class
- A constructor name should be always same as class name.
- A constructor is called at the time of creating an object.
- A constructor is called only once per object.
- A constructor is called and executed automatically

Method

- A method is used for any general purpose tasks like calculations.
- The method name and class name can be same or different.
- A method can be called after creating the object.
- A method can be called any number of times on the object
- A method is executed only when we want it.

29) Explain try and catch with an example.

Ans. The core of exception handling is try and catch . these keywords works together. we cannot have a try without a try.

General form of the try/catch exception handling blocks

```
try
{
    // do risky things
}
catch(Exception ex)
{
    //try to recover
}
```

Try

A try block is simply the keyword try, followed by braces enclosing the risky code that can throw the exception.

```
try
{
}
```

Catch

A catch block consists of the keyword catch followed by a single parameter between parentheses that identify the type of exception that the block is to deal with. this is followed by the code to handle the exception enclosed between braces

```
try{
}
catch(Exception ex){
}
```

Example: To demonstrate arithmetic exception using try-catch block.

```
public class ArithmeticExceptionDemo
{
    public static void main(String args[])
    {
}
```

```

int i=100, j=0;
int k;

try
{
    K=i/j;
}

catch (ArithmetricException e)
{
    System.out.println("exception occurred: divition by zero");
    K=i/(j+2);
}

System.out.println("the value of k is :" + k);
}
}

```

30) Write a short note on Graphics class

- The graphics class provides the framework for all graphics operations within the AWT. It plays two different, but related roles.
- Graphics context: the graphics context is information that will affect drawing operations.
- The graphic context holds the following.
 - a. The component object on which to draw
 - b. Background and foreground colours
 - c. Font
 - d. A translation origin for rendering coordinates
 - e. The region of a component in which graphics can be drawn.
- Graphics class methods: The graphics class provides methods for drawing simple geometric shapes, text and images to the graphics destination. In order to draw, a program requires a valid graphics context. Because the graphics class is an abstract base class, it cannot be instantiated directly. An instance is typically created by components, and handed to the program as an argument to a components update() and paint() methods.
- The graphics context encapsulated by the graphics class is obtained in two ways:
 - a. It is passed to an applet when pain or update method is called.

- b. Abstract classes cannot be instantiated. Therefore, programmers must request a graphics object from a component, which is accomplished by using a method of the form.

Component.getGraphics();

- c. This method is defined in component and returns a graphics object. Therefore, any class that is a subclass of component can be drawn upon. The functionality of the graphics class can be accessed even though it is abstract class.

31) Write program to sort a list of elements in ascending order.

class sorting

{

```
public static void main(String args[])
{
    int a[ ] = new int[5];
    System.out.println(" enter five elements \n");
    for( int i=0;i<5;i++)
        a[i]=Integer.parseInt(args[i]);
    System.out.println("\n before sorting \n");
    for( int i=0;i<5;i++)
        System.out.print(" " +a[i]);
```

bubbleSort(a,5);

```
System.out.println("\n\n after sorting \n");
System.out.println(" Ascending order \n");
for( int i=0;i<5;i++)
    System.out.print(" " +a[i]);
```

}

}

```
private static void bubbleSort(int[ ] arr, int length)
```

{

```
int temp,i,j;
```

```

for(i=0;i<length-1;i++)
{
    If(arr[j] > arr[j+1]
    {
        temp = arr[j];
        arr[j] = arr[j+1];
        arr[j+1] = temp;
    }
}

```

32) Give the steps to create and use a java package with an example.

Ans.

- To create a package, put a package command at the top of a java source file. the classes declare within that file will then belong to the specified package. Since a package defines a namespace, the names of the classes that we put into the file become part of that package's namespace.
- This is the general form of the package statement:

Package pkg;
- Here, pkg is the name of the package. for example, the following statement creates a package called project1.

package pack1;
- Java uses the file system to manage packages, with each package stored in its own directory. for example, the class files for any classes we declare to be part of pack1 must be stored in directory called pack1.
- Step1: create a program which is part of package called pack1

```
package pack1;

public class packageExample
{
    Public void packMethod()
    {
        System.out.println(" I am packMethod() in pack1.packageExample");
    }
}
```

- Step2: the command is `javac -d.packageExample.java`. We can see that `pack1` directory is automatically created by the compiler.
- Step3: create another file called `packageDemo.java` in `pack2` package

```
Package pack2;

Import pack1.*;;

Public class packageDemo
{
    Public static void main(String args[ ])
    {
        packageExample obj=new packageExample();
        obj.packageMethod();
    }
}
```

- In this file, we are trying to import the `pack1` classes and creating an object of `packageExample`

33) Explain steps of executing an applet using simple code.

Ans.

Step1: Writing applet code

Create the applet program called `MyFirstApplet` as shown

```
import java.applet.Applet;
import java.awt.Graphics;

public class MyFirstApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("hello applet", 20, 20);
    }
}
```

Step2: Compile applet code and generate byte code.

Save the applet code and generate the byte code using java compiler javac

Step3: Create an HTML page.

```
<HTML>
<HEAD>
<TITLE> this is my first applet </TITLE>
<BODY>
<applet code=MyFirstApplet.class width= 400 height=500>
</Applet>
</BODY>
<HEAD>
</HTML>
```

Step4: Executing an applet using appletviewer

- There are two ways in which we can run an applet: inside a browser or with a special development tool that displays applets. The tool provided with the standard java JDK is called appletviewer.
- The appletviewer is much easier to use during development.
- Executing an applet using appletviewer
- Appletviewer appletExample.html