

K.S.R. COLLEGE OF ENGINEERING(Autonomous)**Vision of the Institution**

- We envision to achieve status as an excellent educational institution in the global knowledge hub, making self-learners, experts, ethical and responsible engineers, technologists, scientists, managers, administrators and entrepreneurs who will significantly contribute to research and environment friendly sustainable growth of the nation and the world.

Mission of the Institution

- To inculcate in the students self-learning abilities that enable them to become competitive and considerate engineers, technologists, scientists, managers, administrators and entrepreneurs by diligently imparting the best of education, nurturing environmental and social needs.
- To foster and maintain a mutually beneficial partnership with global industries and Institutions through knowledge sharing, collaborative research and innovation.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**Vision of the Department**

- To create ever green professionals for software industry, academicians for knowledge cultivation and researchers for contemporary society modernization.

Mission of the Department

- To produce proficient design, code and system engineers for software development.
- To keep updated contemporary technology and fore coming challenges for welfare of the society.

Programme Educational Objectives (PEOs)

PEO1 : Figure out, formulate, analyze typical problems and develop effective solutions by imparting the idea and principles of science, mathematics, engineering fundamentals and computing.

PEO2 : Competent professionally and successful in their chosen career through life-long learning.

PEO3 : Excel individually or as member of a team in carrying out projects and exhibit social needs and follow professional ethics.

DATE**COURSE FACULTY****H.O.D****PRINCIPAL**

Department of Computer Science and Engineering

Subject Name: Java Programming

Subject Code: 16CS002

Year/Semester: III/V

Course Outcomes: On completion of this course, the student will be able to

- CO1** Understand and apply java programming fundamentals to solve real world problem
- CO2** Understand the concept of overloading and inheritances.
- CO3** Understand the important features of java like packages, interfaces and exception handling.
- CO4** Understand the features of multithreaded programming and I/O operations.
- CO5** Understand the concepts of string manipulations and database connectivity.

Program Outcomes (POs) and Program Specific Outcomes (PSOs)

A. Program Outcomes (POs)

Engineering Graduates will be able to :

- PO1 Engineering knowledge:** Ability to exhibit the knowledge of mathematics, science, engineering fundamentals and programming skills to solve problems in computer science.
- PO2 Problem analysis:** Talent to identify, formulate, analyze and solve complex engineering problems with the knowledge of computer science. .
- PO3 Design/development of solutions:** Capability to design, implement, and evaluate a computer based system, process, component or program to meet desired needs.
- PO4 Conduct investigations of complex problems:** Potential to conduct investigation of complex problems by methods that include appropriate experiments, analysis and synthesis of information in order to reach valid conclusions.
- PO5 Modern tool Usage:** Ability to create, select, and apply appropriate techniques, resources and modern engineering tools to solve complex engineering problems.
- PO6 The engineer and society:** Skill to acquire the broad education necessary to understand the impact of engineering solutions on a global economic, environmental, social, political, ethical, health and safety.
- PO7 Environmental and sustainability:** Ability to understand the impact of the professional engineering solutions in societal and Environmental contexts and demonstrate the knowledge of, and need for sustainable development.
- PO8 Ethics:** Apply ethical principles and commit to professional ethics and responsibility and norms of the engineering practices.
- PO9 Individual and team work:** Ability to function individually as well as on multi-disciplinary teams.
- PO10 Communication:** Ability to communicate effectively in both verbal and written mode to excel in the career.
- PO11 Project management and finance:** Ability to integrate the knowledge of engineering and management principles to work as a member and leader in a team on diverse projects.
- PO12 Life-long learning:** Ability to recognize the need of technological change by independent and life-long learning.

B. Program Specific Outcomes (PSOs)

- PSO1** Develop and Implement computer solutions that accomplish goals to the industry, government or research by exploring new technologies.
- PSO2** Grow intellectually and professionally in the chosen field.

DATE

COURSE FACULTY

H.O.D

PRINCIPAL

JAVA FUNDAMENTALS

TWO MARKS

1. What are the features of java? [Remembering]

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust
7. Architecture neutral
8. Interpreted
9. High Performance
10. Multithreaded
11. Distributed
12. Dynamic

2. Difference between JDK, JRE, and JVM. [Analyzing]

JDK is abbreviated as Java Development Kit which has a physical existence. It can be considered as a kit inside which resides the JRE along with developing tools within it. The programmers and developers mostly use it.

JVM is abbreviated as Java Virtual Machine, is basically a dummy machine or you can say an abstract machine which gives Java programmers a runtime environment for executing the Bytecode. For each execution of your program, the JDK and JRE come into use, and they go within the JVM to run the Java source code.

JRE is abbreviated as Java Runtime Environment, as the name suggests used as a package that gives an environment to run the Java program on your machine

3. What is bytecode? [Remembering]

Bytecode is a highly optimized set of instructions designed to be executed by the java run-time system. Which is called the java virtual machine (JVM). JVM is an interpreter for bytecode.

4. What is a variable? What are the different types of variables? [Remembering]

Variable are locations in the memory that can hold values. Java has three kinds of variable namely,

Instance variable

Local variable

Class variable

Local variables are used inside blocks as counts or in methods as temporary variables. Once the block or the method is executed, the variable ceases to exist. Instance variable are used to define attributes or the state of a particular object. These are used to store information needed by multiple methods in the objects.

5. What are the difference between static variable and instance variable? [Analyzing]

The data or variables, defined within a class are called instance variables.

Instance variables declared as static are, essentially, global variables. When objects of its class are declared, no copy of a static variable is made.

6. What are primitive datatypes in java? [Remembering]

There are 8 types of primitive data types:

1. boolean data type

2. byte data type
3. char data type
4. short data type
5. int data type
6. long data type
7. float data type
8. double data type

7. Define Array? How to declare an array? [Remembering]

Java array is an object which contains elements of a similar data type. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

There are two types of array.

1. Single Dimensional Array
 2. Multidimensional Array
- int a[]=new int[5];**//declaration

8. List out the operator in Java. [Creating]

1. Arithmetic Operators
2. Increment and Decrement Operators
3. Bitwise Operators
4. Relational Operators
5. Logical Operators
6. Assignment Operators

9.what is object? [Creating]

An object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

1. An object is *a real-world entity*.
2. An object is *a runtime entity*.
3. The object is *an entity which has state and behavior*.
4. The object is *an instance of a class*.

10.Define Construcor? [Remembering]

Constructor in java is a special type of method that is used to initialize the object. Java constructor is invoked at the time of object creation. It constructs the values i.e. provides data for the object that is why it is known as constructor.

11. What are the Types of java constructors? [Understanding]

There are two types of constructors: 1. Default constructor (no-arg constructor) 2. Parameterized constructor

12.What is meant by garbage collection? [Remembering]

It frees memory allocated to objects that are not being used by the program any more - hence the name "garbage".In certain languages like C++, dynamically allocated objects must be manually released by use of a delete operator. In Java deallocation happens automatically. The technique that accomplishes this is called garbage collection.

13.Define method? [Creating]

Methods are functions that operates on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Just as there are class and instance variable, there are class and instance methods. Instance methods apply and operate on an instance of the class while class methods operate on the class.

14. What is the use of 'this' Keyword? [Analyzing]

1. this can be used to refer current class instance variable.
2. this can be used to invoke current class method (implicitly)
3. this() can be used to invoke current class constructor.
4. this can be passed as an argument in the method call.
5. this can be passed as argument in the constructor call.

6. this can be used to return the current class instance from the method.

15. What gives java it's "write once and run anywhere" nature? [Understanding]

All Java programs are compiled into class files that contain byte codes. These byte codes can be run in any platform and hence java is said to be platform independent.

16. Why Java is Platform Independent. [Creating]

Java is platform independent because it can run on any platform i.e on windows,mac etc. ...

Because it contains its own virtual machine which have JRE which runs java program as byte code. You may have different JRE for different platforms .Once you installed it in your system you can run any java application on it.

17. Why Java is Architecture-Neutral. [Analyzing]

Java was designed to support applications on networks. To enable a Java application to execute anywhere on the network, the compiler generates an architecture-neutral object file format--the compiled code is executable on many processors, given the presence of the Java runtime system

18 .What is finalize() method? [Remembering]

Finalize () method is used just before an object is destroyed and can be called just prior to garbage collection

16 Marks

1. Explain about the Constructors and its type with example program. [Remembering]
2. Write short notes on Arrays with its declaration and with its type. Explain with suitable Program. [Creating]
3. Explain briefly about the features of java. [Analyzing]
4. List out the operator in Java.write Program and and explain it. [Applying]
5. What is garbage collection.Explain role of Garbage Collection in java[Remembering]

Unit-2

Method Overloading and Inheritance

TWO MARKS

1. Define method overloading? [Remembering]

In Java it is possible to define two or more methods within the same class that share the same name, as long as their parameter declarations are different. When this is the case, the methods are said to be *overload*, and the process is referred to as *method overloading*.

2. What is meant by an innerclass? .[Applying]

An inner class is a nested class whose instance exists within an instance of its enclosing class and has direct access to the instance members of its enclosing instance

```
class <EnclosingClass>
{
    class <InnerClass>
    {
        }
    }
}
```

3. What are the uses of the keyword 'final'? [Remembering]

- The class can be declared as final, if instances or subclasses are not to be created.
- The variables are declared as final, value of the variable must be provided at the time of declaration.
- The Method can be declared as final indicating that they cannot be overridden by subclasses.

4. What are static methods? [Remembering]

Static methods and variables can be used independently of any object. To do so, you need only specify the name of their class following by the dot operator.

5. What is inheritance? [Applying]

In Object-Oriented programming, inheritance refers to the properties of a class being available to many other classes. A derived class / sub class is one that has been created from an existing class. Inheritance is the process of deriving a class from a super class or a base class. No changes are made to the base class. The derived class has a larger set of properties than its base class. Inheritance has two advantages

- a) Reusability of code
- b) Data and methods of a super class are physically available to its subclasses

6. What is the use of 'Super' Keyword? Give an example. [Remembering]

Usage of 'super' keyword

1. The first calls the superclass constructor
2. To access a member of the superclass that has been hidden by a member of a subclass

7. Difference between method overloading and method overriding in java. [Analyzing]

Overloading	Overriding
Whenever same method or Constructor is existing multiple times within a class either with different number of parameter or with different type of parameter or with different order of parameter is known as Overloading.	Whenever same method name is existing multiple time in both base and derived class with same number of parameter or same type of parameter or same order of parameters is known as Overriding.
Arguments of method must be different at least arguments.	Argument of method must be same including order.
Method signature must be different.	Method signature must be same.
Private, static and final methods can be overloaded.	Private, static and final methods can not be override.
Access modifiers point of view no restriction.	Access modifiers point of view not reduced scope of Access modifiers but increased.

8. What is constructor overloading. [Understanding]

Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

9. How to pass object as argument in java? [Remembering]

1. We can pass Object of any class as parameter to a method in java.
2. We can access the instance variables of the object passed inside the called method.
3. It is good practice to initialize instance variables of an object before passing object as parameter to method otherwise it will take default initial values.

10. What is recursion? [Understanding]

Java supports recursion. Recursion is the process of defining something in terms of itself. As it relates to java programming, recursion is the attribute that allows a method to call itself. A method that calls itself is said to be recursive.

11. List Access Specifiers in java. [Analyzing]

Java Access Specifiers (also known as Visibility Specifiers) regulate access to classes, fields and methods in Java.

There are 4 types of java access modifiers:

1. private
2. default
3. protected
4. public

12. What are the uses of the keyword 'final'? [Analyzing]

- 1 The class can be declared as final, if instances or subclasses are not to be created.
- 2 The variables are declared as final, value of the variable must be provided at the time of declaration.
- 3 The Method can be declared as final indicating that they cannot be overridden by subclasses.

13. What are static methods? [Applying]

Static methods and variables can be used independently of any object. To do so, you need only specify the name of their class following by the dot operator.

14. What is inheritance? [Understanding]

In Object-Oriented programming, inheritance refers to the properties of a class being available to many other classes. A derived class / sub class is one that has been created from an existing class. Inheritance is the process of deriving a class from a super class or a base class. No changes are made to the base class. The derived class has a larger set of properties than its base class. Inheritance has two advantages

- a) Reusability of code
- b) Data and methods of a super class are physically available to its subclasses

15. Define abstract class? [Analyzing]

Abstract classes are classes from which instances are usually not created. It is basically used to contain common characteristics of its derived classes. Abstract classes are generally higher up the hierarchy and act as super classes. Methods can also be declared as abstract. This implies that non-abstract classes must implement these .

16. What are the rules for method overriding. [Analyzing]

1. The method must have the same name as in the parent class
2. The method must have the same parameter as in the parent class.
3. There must be an IS-A relationship (inheritance).

17. What is Static Polymorphism? [Applying]

Static Polymorphism is also known as compile time binding or early binding. 2. Static binding happens at compile time. Method overloading is an example of static binding where binding of method call to its definition happens at Compile time.

18. What are the 3 ways of Overloading Method? [Creating]

1. Different Number of parameters in argument list.
2. Difference in data type of arguments.
3. Sequence of data type of arguments

19. What are the types of inheritance [Applying]

1. Single Inheritance
2. Multiple Inheritance (Through Interface)
3. Multilevel Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance (**Through Interface**).

20. Can you use this () and super () both in a constructor? [Analyzing]

We cannot have two statements as first statement, so we either need to call super() or this() but not the both. We can't use both the keywords in the constructor. In Java there is a rule that this() and super() must be first statement in the constructor. So we can't use both together in a constructor.

21. How can we achieve multiple inheritance in java? [Creating]

Way to achieve multiple inheritance through interfaces in java. class A extends B, C // this is not possible in java directly but can be achieved indirectly. We can implement both of these using the code below... We CANNOT extend two objects, but we can implement two interfaces.

22 What is meant by Nested Class? [Understanding]

Class within another class, such classes are known as *nested* classes.

Nested classes are divided into two categories:

1. **static nested class** : Nested classes that are declared *static* are called static nested classes.
2. **inner class** : An inner class is a non-static nested class.

16Marks

1. Explain method overloading with example program. Write the three ways of overloading method. [Analyzing]
2. Explain Constructor Overloading with example program. [Understanding]
3. Discuss about access specifier in java. Write a program use all access specifier. [Creating]
4. Write about Method overriding .Write Program for addition of two number . [Applying]
5. Explain about Inheritance .List out types of Inheritance. Write a java Program to explain all type of inheritance. [Analyzing]

UNIT-3

Package, Interface and Exception handling

2 Marks

1. What is a package? [Remembering]

Packages contain a set of classes in order to ensure that class names are unique. Packages are containers for classes that are used to compartmentalize the class name space. Packages are stored in a hierarchical manner and are explicitly imported into new class definition. A period is used as separator to enable this.

2. Write a note on import statement? [Applying]

Classes external to a program be imported before they can be used. To import a class the *import* keyword should be used as given below

```
import <classname>
```

The classes in Java are arranged in hierarchical order. The Java library consists of a number of package. These package contain a set of related classes. The whole path of the class must be specified to import a class from the Java library, For instance, to import the Date class from the **util** package use the following code.

```
import java.util.Date;
```

It is also possible to import all classes that belong to a package using the * symbol.

3.Explain the usage of Java packages [Remembering]

This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the nonauthorized classes.

4. Define interface? [Understanding]

An interface is a collection of abstract behavior that individual classes can implement. It is defined like a class. An interface consists of a set of method definition. Any class implementing it should provide code for all its methods.

5 What is the useful of Interfaces? [Applying]

- a) Declaring methods that one or more classes are expected to implement
- b) Capturing similarities between unrelated classes without forcing a class relationship.
- c) Determining an object's programming interface without revealing the actual body of the class.

6. Define an exception.[Understanding]

An exception is an abnormal condition, which occurs during the execution of a program Exceptions are erroneous events like division by zero, opening of a file that does not exist, etc. A java execution is an object, which describes the error condition that has materialized in the program.

7. How to access package from another package? [Remembering]

There are three ways to access the package from outside the package.

- 1. `import package.*;`
- 2. `import package.classname;`
- 3. fully qualified name.

8. What are the types of Exception in Java. [Remembering]

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception.

- 1. Checked Exception
- 2. Unchecked Exception.

9. Difference between Checked and Unchecked Exceptions [Analyzing]

1) Checked Exception

The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes which inherit RuntimeException are known as unchecked exceptions e.g.

ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime

10.What are the keyword in Exception. [Applying]

- 1. `try`
- 2. `catch`
- 3. `throw`
- 4. `throws`
- 5. `finally`

11. What is use of try block? [Remembering]

The "try" keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.

12. What is catch block? [Remembering]

The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.

13. What is use of 'throw statement' give an example? (or) state the purpose of the throw statement. [Remembering]

Whenever a program does not want to handle exception using the try block, it can use the throws clause. The throws clause is responsible to handle the different types of exceptions generated by the program. This clause usually contains a list of the various types of exception that are likely to occur in the program.

14. What is throws in java? [Remembering]

The Java throws keyword is used to declare an exception. It gives an information to the programmer that there may occur an exception so it is better for the programmer to provide the exception handling code so that normal flow can be maintained.

```
return_type method_name() throws exception_class_name{  
    //method code }
```

15. Difference between throw and throws [Analyzing]

s.no	throw	throws
1	Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
2	Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.

16. What is finally in Java with example? [Analyzing]

Java Finally Block Examples. The finally block always executes immediately after try-catch block exits. The finally block is executed incase even if an unexpected exception occurs. ... The runtime system always executes the code within the finally block regardless of what happens in the try block.

17. What is the difference between error and exception? [Remembering]

An Error "indicates serious problems that a reasonable application should not try to catch."

An Exception "indicates conditions that a reasonable application might want to catch." Error along with RuntimeException & their subclasses are unchecked exceptions. All other Exception classes are checked exceptions.

18. How does nested try catch work in Java? [Analyzing]

Nested try catch blocks. Exception handlers can be nested within one another. A try, catch or a finally block can in turn contains another set of try catch finally sequence. In such a scenario, when a particular catch block is unable to handle an Exception, this exception is rethrown.

16marks

1. Define Interfaces? Explain the extension of interfaces, implementation and accessing it. . [Analyzing]
2. What are the Different Exceptions caught, Explain the types with example. [Creating]
3. Explain about Exception handling in Java with example. [Remembering]
4. Explain try, catch and finally statements with example. [Analyzing]
5. Define package and its use. Explain about types of package and how to import package in another package with example program. [Analyzing]

UNIT-4**Multithreaded Programming and I/O Operations****2marks****1. How to create Thread. [Analyzing]**

There are two ways to create a thread:

1. By extending Thread class
2. By implementing Runnable interface.

2. What are the different states of a thread? [Remembering]

The different thread states are ready, running, waiting and dead.

3. Why are there separate wait and sleep methods? [Creating]

Sleep (long) method maintains control of thread execution but delays the next action until the sleep time expires. The wait method gives up control over thread execution indefinitely so that other threads can run.

4. What is thread? [Creating]

A thread is a single sequential flow of control within a program. A single thread also has a beginning, an end, a sequence, and at any given time during the runtime of the thread there is a single point of execution. However, a thread itself is not a program. It cannot run on its own, but runs within a program.

5.What are thread states? [Remembering]

NEW A thread that has not yet started is in this state. RUNNABLE A thread executing in the Java virtual machine is in this state. BLOCKED A thread that is blocked waiting for a monitor lock is in this state. WAITING A thread that is waiting indefinitely for another thread to perform a particular action is in this state. TIMED WAITING A thread that is waiting for another thread to perform an action for up to a specified waiting time is in this state.

6. What is difference between starting thread with Run () and start () method? [Evaluating]

Main difference is that when program calls start() method a new Thread is created and code inside run() method is executed in new Thread while if you call run() method directly no new Thread is created and code inside run() will execute on current Thread.

7. Can we call run method directly? [Remembering]

No, you can not directly call run method to start a thread. You need to call start method to create a new thread. If you call run method directly , it won't create a new thread and it will be in same stack as main. As you can see when we are directly calling run method, it is not creating new threads.

8. Define multithreading? [Evaluating]

A thread is a line of execution. It is the smallest unit of code that is dispatched by the scheduler. Thus, a process can contain multiple threads to execute its different sections. This is called multithread.

9.What is the need of Thread Priorities? [Remembering]

Thread priorities are used by the thread scheduler to decide when each thread be allowed to run. Higher-priority threads get more CPU time than lower-priority threads. To set a thread's priority, use the setPriority() method, which is a member of Thread. final void setPriority(int level).

10. What are three ways in which a thread can enter the waiting state? [Analyzing]

A thread can enter the waiting state by invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

11.Write about thread priority. [Remembering]

Each thread have a priority. Priorities are represented by a number between 1 and 10.

3 constants defined in Thread class:

1. public static int MIN_PRIORITY
2. public static int NORM_PRIORITY
3. public static int MAX_PRIORITY

Default priority-5

MIN_PRIORITY-1

MAX_PRIORITY-10

12. What is synchronized keyword? In what situations you will Use it? [Evaluating]

Synchronization is the act of serializing access to critical sections of code. We will use this keyword when we expect multiple threads to access/modify the same data. To understand synchronization we need to look into thread execution manner.

13. What is Inter-thread communication? [Remembering]

Inter-thread communication or Co-operation is all about allowing synchronized threads to communicate with each other. It is implemented by following methods of Object class:

1. wait()
2. notify()
3. notifyAll()

14. What are Byte Stream in Java? [Analyzing]

The byte stream classes provide a rich environment for handling byte-oriented I/O.

List of Byte Stream classes

1. ByteArrayInputStream
2. ByteArrayOutputStream
3. FilteredByteStreams
4. BufferedByteStreams

15. What are Character Stream in Java? [Remembering]

The Character Stream classes provide a rich environment for handling character-oriented I/O.

List of Character Stream classes

1. FileReader
2. FileWriter
3. CharArrayReader
4. CharArrayWriter

16. What is Java Streaming? [Analyzing]

Java streaming is nothing more than a flow of data. There are input streams that direct data from the outside world from the keyboard, or a file for instance, into the computer; and output streams that direct data toward output devices such as the computer screen or a file.

17. Write note on FileInputStream class. [Remembering]

The FileInputStream class creates an InputStream that you can use to read bytes from a file. Its two most common constructors are FileInputStream(String filepath) FileInputStream(File fileobj)

18. Write note on FileOutputStream class. [Remembering]

FileOutputStream creates an OutputStream that you can use to write bytes to a file. Its most commonly used constructors are FileOutputStream(String filepath) FileOutputStream(File fileobj) FileOutputStream(String filepath, Boolean append).

\\ They can throw an IOException or a SecurityException.

19.what is PrintwriterClass[Remembering]

Java PrintWriter class is the implementation of Writer class. It is used to print the formatted representation of objects to the text-output stream.

20.Write about Scanner Class[Analyzing]

Java Scanner class comes under the java.util package. Java has various ways to read input from the keyboard, the java.util.Scanner class is one of them.

The Java Scanner class breaks the input into tokens using a delimiter that is whitespace by default. It provides many methods to read and parse various primitive values.

Java Scanner class is widely used to parse text for string and primitive types using a regular expression.

16Marks

1. Explain in detail about thread. Write about two ways of creating thread with suitable program. [Remembering]
2. Write a Java program that prints numbers from 1 to 10 line by line after every 5 seconds. [Analyzing]
3. What is thread synchronization? Discuss with an example. [Remembering]
4. Write a Java program for creating four threads to perform the following operations i) Getting N numbers as input ii) Printing the even numbers iii) Printing the odd numbers iv) Computing the average [Analyzing]
5. With neat diagram discuss about life cycle of thread and its priority. [Evaluating]
6. Write a java program to read a file and copy content of one file into another file. [Analyzing]
7. Discuss about scanner class with an example program. [Remembering]

Unit V

String and Database connectivity

1. What is String in Java ? Is String is data type? [Remembering]

String in Java is not a primitive data type like int, long or double. String is a class or in more simple term a user defined type. This is confusing for some one who comes from C background. String is defined in java.lang package and wrappers its content in a character array. String provides equals() method to compare two String and provides various other method to operate on String like toUpperCase() to convert String into upper case, replace() to replace String contents, substring() to get substring, split() to split long String into multiple String.

2. Why String is final in Java [Remembering]

String is final by design in Java, some of the points which makes sense why String is final is Security, optimization and to maintain pool of String in Java. for details on each of this point

3 What is Difference between String and String Buffer in Java [Remembering]

This is probably the most common question on String I have seen in Java interviews. Though String and String buffer are two different class they are used in context of concatenating two Strings, Since String is immutable in Java every operation which changes String produces new String, which can be avoided by using String buffer. See String vs String Buffer for more details.

4. What is difference in String on C and Java [Remembering]

If you have mentioned C in your resume, then you are likely to face this String interview question. Well C String and Java String are completely different to each other, C String is a null terminated character array while String in Java is an Object. Also String is more feature rich in Java than C.

5. Does String is thread-safe in Java [Understanding]

If you are familiar with the concept of immutability and thread-safety you can easily answer this String interview question in Java. Since String is immutable, it is thread-safe and it can be shared between multiple thread without external synchronization.

6 What is mean by String Literals? [Understanding]

The string of characters is represented as String literals in Java. In Java a string is not a basic data type, rather it is an object. These strings are not stored in arrays as in C language. There are few methods provided in Java to combine strings, modify strings and to know whether to strings have the same value.

Lets see some more examples of string literals:

```
"" // the empty string
```

```
"\" // a string containing "
```

```
"This is a string" // a string containing 16 characters
```

```
"This is a " + // actually a string-valued constant expression,
```

```
"two-line string" // formed from two string literals
```

7. List the two methods to search a string. [Analyzing]

1. indexOf()

2. lastIndexOf()

8. Define String Buffer Class. [Understanding]

String Buffer class is a mutable class unlike the String class which is immutable. Both the capacity and character string of a String Buffer Class. String Buffer can be changed dynamically. String buffers are preferred when heavy modification of character strings is involved (appending, inserting, deleting, modifying etc).

Strings can be obtained from string buffers. Since the String Buffer class does not override the equals() method from the Object class, contents of string buffers should be converted to String

objects for string comparison. A String Index Out Of Bounds Exception is thrown if an index is not valid when using wrong index in String Buffer manipulations

It consist of the following functions append, insert, reverse, set CharAt, charAt, length, delete CharAt, substring, delete, capacity.

9. What is the use of to String() Method? [Analyzing]

The method is used to get a String object representing the value of the Number Object.

If the method takes a primitive data type as an argument then the String object representing the primitive data type value is return. If the method takes two arguments then a String representation of the first argument in the radix specified by the second argument will be returned

10. Explain different types of jdbc drivers. [Remembering]

There are four types of drivers available.

Type-1 Driver(Jdbc Odbc bridge) :

Type-2 Driver (Native Driver) :

Type-3 Driver (Network Driver) :

Type-4 Driver(Pure java Driver) :

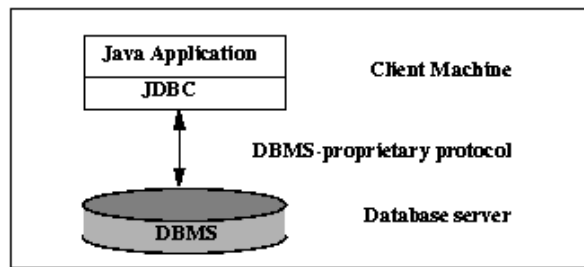
11. Write steps to connect to database [Analyzing]

- Load the driver
- Create Connection object
- Create Statement object
- Execute Query
- Process Data
- Close the Connection

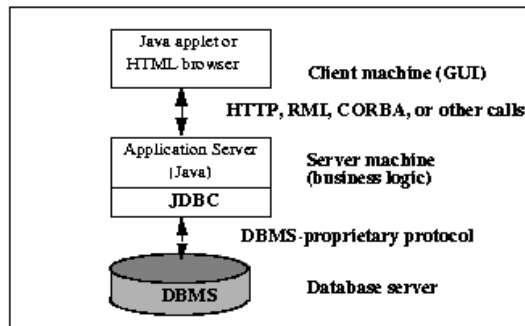
12. List JDBC Product Components [Analyzing]

- The JDBC API
- JDBC Driver Manager
- JDBC Test Suite
- JDBC-ODBC Bridge

13. What is two tier architecture? [Analyzing]



14. What is three tier architecture? [Analyzing]



15. Explain Type-1 driver [Analyzing]

Type-1 driver or JDBC-ODBC bridge driver uses ODBC driver to connect to the database. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC function calls. Type-1 driver is also called Universal driver because it can be used to connect to any of the databases.

- As a common driver is used in order to interact with different databases, the data transferred through this driver is not so secured.
- The ODBC bridge driver is needed to be installed in individual client machines.
- Type-1 driver isn't written in java, that's why it isn't a portable driver.

16. Explain Type-2 driver. [Remembering]

The Native API driver uses the client -side libraries of the database. This driver converts JDBC method calls into native calls of the database API. In order to interact with different database, this driver needs their local API, that's why data transfer is much more secure as compared to type-1 driver.

- Driver needs to be installed separately in individual client machines
- The Vendor client library needs to be installed on client machine.
- Type-2 driver isn't written in java, that's why it isn't a portable driver

17. Explain Type-3 driver [Analyzing]

The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol. Here all the database connectivity drivers are present in a single server, hence no need of individual client-side installation.

- Type-3 drivers are fully written in Java, hence they are portable drivers.
- No client side library is required because of application server that can perform many tasks like auditing, load balancing, logging etc.
- Network support is required on client machine.
- Maintenance of Network Protocol driver becomes costly because it requires database-specific coding to be done in the middle tier.

18. Explain Type-4 driver. [Analyzing]

Type-4 driver is also called native protocol driver. This driver interact directly with database. It does not require any native database library, that is why it is also known as Thin Driver.

- Does not require any native library and Middleware server, so no client-side or server-side installation.
- It is fully written in Java language, hence they are portable drivers.

16 marks

- 1 Explain about character extraction methods available in java.[Analyzing]
- 2 Discuss about JDBC architecture and explain in detail [Creating]s
- 3 List out five steps in JDBC connection establishment with example program [Remembering]
- 4 Elaborate String and its methods and StringBuffer and its methods with suitable example. [Understanding]
- 5 Explain the different types of drivers in detail [Analyzing]