

DMI COLLEGE OF ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY
CS3391- OBJECT ORIENTED PROGRAMMING
Unit 1
QUESTION BANK WITH ANSWER

UNIT I INTRODUCTION TO OOP AND JAVA

9

Overview of OOP – Object oriented programming paradigms – Features of Object Oriented Programming – Java Buzzwords
– Overview of Java – Data Types, Variables and Arrays – Operators – Control Statements – Programming Structures in Java
– Defining classes in Java – Constructors-Methods -Access specifiers - Static members- JavaDoc comments

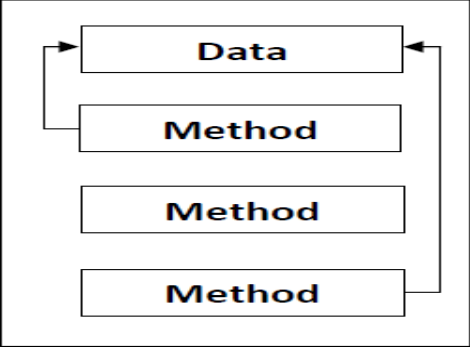
UNIT-I/ PART-A

1	<p>What are the OOP Principles?</p> <p>The principles of object-oriented programming is Class inheritance , interface implementation, abstraction of data and behavior, encapsulation of data and class implementation, polymorphism and virtual methods.</p>
2	<p>What are the four cornerstones of OOP?</p> <p><i>Abstraction:</i> Can manage complexity through abstraction. Gives the complete overview of a particular task and the details are handled by its derived classes. Example : Car.</p> <p><i>Encapsulation:</i> Nothing but data hiding, like the variables declared under private of a particular class is accessed only in that class and cannot access in any other the class.</p> <p><i>Inheritance:</i> Is the process in which one object acquires the properties of another object, ie., derived object.</p> <p><i>Polymorphism:</i> One method different forms, ie., method overriding and interfaces are the examples of polymorphism.</p>
3	<p>What are the features of Object Oriented Programming?</p> <ul style="list-style-type: none"> • Emphasis is on data rather than procedure. • Programs are divided into objects. • Data structures are designed such that they characterize the objects. • Functions that operate on the data of an object are tied together. • Data is hidden and cannot be accessed by external functions. • Objects may communicate with each other through functions. • New data and functions can easily be added whenever necessary. • Follows bottom-up approach.
4	<p>What are the features of Java Language?</p> <p>The features of Java Language are Simple, Object-Oriented, Portable, Platform independent, Secured, Robust, Architecture neutral, Dynamic, Interpreted, High Performance, Multithreaded and Distributed.</p>
5	<p>How Java supports platform independency?</p> <ul style="list-style-type: none"> • The meaning of platform independent is that, the java source code can run on all operating systems. a compiler is a program that translates the source code for another program from a programming language into executable code. • This executable code may be a sequence of machine instructions that can be executed by the CPU directly, or it may be an intermediate representation that is interpreted by a virtual machine. This intermediate representation in Java is the Java Byte Code. • It is the Bytecode that makes it platform independent. This adds to an important feature in the JAVA language termed as portability. Every system has its own JVM which gets installed automatically when the jdk software is installed. For every operating system separate JVM is available which is capable to read the .class file or byte code. An important point to be noted is that while JAVA is platform-independent language, the JVM is platform-dependent
6	<p>Give the contents of Java Environment (JDK).</p> <p>The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets.</p> <p>It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.</p>
7	<p>What is Java Interpreter?</p>

	It is a Java Virtual Machine. An <i>interpreter</i> is a program that reads in as input a source program, along with data for the program, and translates the source program instruction by instruction. For example, the Java interpreter java translate a .class file into code that can be executed natively on the underlying machine.	
8	Give any 4 differences between C and Java.	
	C	Java
	C is a procedural Language	Java is Object Oriented Language
	C is a compiled language.	Java is an Interpreted language
	C uses the top-down {sharp & smooth} approach	JAVA uses the bottom-up {or the rocks} approach.
	C does not support overloading	JAVA supports Method Overloading
9	Give any 4 differences between C++ and Java.	
	C++	Java
	C++ generates object code and the same code may not run on different platforms.	Java is interpreted for the most part and hence platform independent.
	C++ supports structures, unions, templates, operator overloading, pointers and pointer arithmetic.	Java does not support pointers, templates, unions, operator overloading, structures etc.
	C++ support destructors, which is automatically invoked when the object is destroyed.	Java support automatic garbage collection. It does not support destructors as C++ does.
10	Distinguish between procedure oriented programming (POP) and Object oriented programming.(OOP)	
	POP	OOP
	In POP, program is divided into small parts called functions.	In OOP, program is divided into parts called objects.
	POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.
	POP does not have any proper way for hiding data so it is less secure.	OOP provides Data Hiding so provides more security.
	In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
	In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system. Example of POP are : C, VB, FORTRAN, Pascal	In OOP, data cannot move easily from function to function, it can be kept public or private so we can control the access of data. Example of OOP are : C++, JAVA, VB.NET, C#.NET.
11	What are the data types supported in Java? Operator in java is a symbol that is used to perform operations like +, -, *, / etc. There are many types of operators in java which are Unary Operator, Arithmetic Operator, Shift Operator, Relational Operator, Bitwise Operator, Logical Operator, Ternary Operator and, Assignment Operator.	
12	Define Abstraction. Abstraction refers to the act of representing the essential features without including the background details or explanations. It reduces the complexity and increases the efficiency. Small programs can be easily upgraded to large programs. Software complexity can easily be managed.	
13	What is Polymorphism? Polymorphism is the ability to take more than one form and refers to an operation exhibiting	

different behavior instances. Object oriented programs use polymorphism to carry out the same operation in a manner customized to the object. It allows a single name/operator to be associated with different operation depending on the type of data passed to it.

14	<p>Java Language is platform independent . Justify your answer (Dec 2022)</p> <p>Yes. Java is a platform-independent language, meaning we run the same code on multiple platforms. Java achieves this using JVM and Byte Code. Java compiler converts the programming code into byte code. Byte code is platform-independent and can be run on any processor or system.</p>																												
15	<p>Difference between Method and Constructor.(Dec 2022)</p> <table> <tr> <th>Sr. No.</th><th>Constructor</th><th>Method</th></tr> <tr> <td>1.</td><td>A block of code that initialize at the time of creating a new object of the class is called constructor.</td><td>A set of statements that performs specific task with and without returning value to the caller is known as method.</td></tr> <tr> <td>2.</td><td>It is mainly used for initializing the object.</td><td>It is mainly used to reuse the code without writing the code again.</td></tr> <tr> <td>3.</td><td>It is implicitly invoked by the system.</td><td>A method is called by the programmer.</td></tr> <tr> <td>4.</td><td>The new keyword plays an important role in invoking the constructor.</td><td>Method calls are responsible for invoking methods.</td></tr> <tr> <td>5.</td><td>It has no return type. It can or cannot return any value to the caller.</td><td>So, it has a return type.</td></tr> <tr> <td>6.</td><td>The constructor name will always be the same as the class name.</td><td>We can use any name for the method name, such as addRow, addNum and subNumbers etc.</td></tr> <tr> <td>7.</td><td>A class can have more than one parameterized constructor. But constructors should have different parameters.</td><td>A class can also have more than one method with the same name but different in arguments and datatypes.</td></tr> <tr> <td>8.</td><td>Sub-class cannot inherit parent class constructor.</td><td>Sub-class can inherit the method of the parent class.</td></tr> </table>		Sr. No.	Constructor	Method	1.	A block of code that initialize at the time of creating a new object of the class is called constructor.	A set of statements that performs specific task with and without returning value to the caller is known as method.	2.	It is mainly used for initializing the object.	It is mainly used to reuse the code without writing the code again.	3.	It is implicitly invoked by the system.	A method is called by the programmer.	4.	The new keyword plays an important role in invoking the constructor.	Method calls are responsible for invoking methods.	5.	It has no return type. It can or cannot return any value to the caller.	So, it has a return type.	6.	The constructor name will always be the same as the class name.	We can use any name for the method name, such as addRow, addNum and subNumbers etc.	7.	A class can have more than one parameterized constructor. But constructors should have different parameters.	A class can also have more than one method with the same name but different in arguments and datatypes.	8.	Sub-class cannot inherit parent class constructor.	Sub-class can inherit the method of the parent class.
Sr. No.	Constructor	Method																											
1.	A block of code that initialize at the time of creating a new object of the class is called constructor.	A set of statements that performs specific task with and without returning value to the caller is known as method.																											
2.	It is mainly used for initializing the object.	It is mainly used to reuse the code without writing the code again.																											
3.	It is implicitly invoked by the system.	A method is called by the programmer.																											
4.	The new keyword plays an important role in invoking the constructor.	Method calls are responsible for invoking methods.																											
5.	It has no return type. It can or cannot return any value to the caller.	So, it has a return type.																											
6.	The constructor name will always be the same as the class name.	We can use any name for the method name, such as addRow, addNum and subNumbers etc.																											
7.	A class can have more than one parameterized constructor. But constructors should have different parameters.	A class can also have more than one method with the same name but different in arguments and datatypes.																											
8.	Sub-class cannot inherit parent class constructor.	Sub-class can inherit the method of the parent class.																											
16	<p>Define Objects and Classes in Java(May 2019)</p> <ul style="list-style-type: none"> Class is a collection of data and the function that manipulate the data. The data components of the class are called data fields and the function components of the class are called member functions or methods. The class that contains main function is called main class. Object is an instance of a class. The objects represent real world entity. The objects are used to provide a practical basis for the real world. Objects are used to understand the real world. The object can be declared by specifying the name of the class. 																												

17	<p>Write the syntax for declaration of class and creation of objects?</p> <p>A class is declared using class keyword. A class contains both data and method that operate on that data. Thus, the instance variables and methods are known as class members. When creating an object from a class</p> <p>Declaration – A variable declaration with a variable name with an object type.</p> <p>Instantiation – The 'new' keyword is used to create the object.</p> <p>Initialization – The 'new' keyword is followed by a call to a constructor. This call initializes the new object.</p> <p>class Student</p> <pre>{ String name; int rollno; int age; }</pre> <p>Student std=new Student();</p> <ul style="list-style-type: none"> • std is instance/object of Student class. • new keyword creates an actual physical copy of the object and assign it to the std variable. • The new operator dynamically allocates memory for an object.
18	<p>Define Encapsulation (Apr/May 2012) (Apr 2017)</p> <p>The wrapping up of data and functions into a single unit is known as data encapsulation. Here the data is not accessible to the outside the class. The data inside that class is accessible by the function in the same class. It is normally not accessible from the outside of the component.</p> <p style="text-align: center;">Class</p> 

19	<p>What is Inheritance? What are its types?(Nov 2018)</p> <ul style="list-style-type: none"> • Inheritance is a mechanism of reusing the properties and extending existing classes without modifying them, thus producing hierarchical relationships between them. • Inheritance is a property by which the new classes are created using the old classes. • The old classes are referred as base classes and the new classes are referred as derived classes. That means the derived classes inherit the properties of base class. • extends and implements keywords are used to describe inheritance in Java. <p>Types of inheritance are: Single inheritance, Multi-level inheritance, Hierarchical inheritance, Hybrid inheritance.</p> <p>Syntax : class Subclass-name extends Superclass-name { //methods and fields }</p>
20	<p>Define class[NOV/DEC 2011] Class is a template for a set of objects that share a common structure and a common behavior.</p>
21	<p>What do you mean by Dynamic Initialization?(Nov 2016) Java is a flexible programming language which allows the dynamic initialization of variables. In other words, at the time of declaration one can initialize the variables. In java we can declare the variable at any place before it is used. Example: int a=10; float d=2.34f;</p>
22	<p>What do you mean by Variable? What are the rules for variable declaration?(Nov 2017) Variable is a fundamental unit of storage in java. The variables are used in combination with identifiers, data types, operators and some value for initialization. The syntax of variable declaration will be: data_type name_of_variable[=initialization];</p>
23	<p>What are the steps for execution of a java program? A program is written in JAVA, the javac compiles it. The result of the JAVA compiler is the .class file or the bytecode and not the machine native code (unlike C compiler). The bytecode generated is a non-executable code and needs an interpreter to execute on a machine. This interpreter is the JVM and thus the Bytecode is executed by the JVM. And finally program runs to give the desired output.</p> <pre> graph LR A[Java Source Program] --> B[Java Compiler (javac)] B --> C[Java Bytecode] C --> D[Network or .fileSystem] D --> E[Java Bytecode] E --> F[Bytecode Verifier] F --> G[JVM] F --> H[JIT compiler] G --> I[Operating System] H --> I </pre>
24	<p>What do you mean by Bytecode? What is JVM and JIT?(May 2019) Bytecode is an intermediate form of java programs. We get bytecode after compiling the java program using a compiler called javac. The bytecode is to be executed by java runtime environment which is called as Java Virtual Machine(JVM). The programs that are running on JVM must be compiled into a binary format which is denoted by .class files. The JVM executes .class or .jar files, by either interpreting it or using a just-in-time compiler (JIT). The JIT is used for compiling and not for interpreting the file. It is used in most JVMs today to achieve greater speed.</p>
25	<p>What is difference between Methods and Constructor?(May 2017) A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator. A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.</p>
26	<p>What are the different datatypes in java?</p>

	TYPE	SIZE	RANGE	SYNTAX
	byte	8 bits	-128 to 127	byte i,j;
	short	16 bits	-32768 to 32767	short a,b;
	int	32 bits	-2,147,483,648 to 2,147,483,647	int i,j;
	long	64 bits	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	long x,y;
	float	32 bits	1.4e-045 to 3.4e+038	float p,q;
	double	64 bits	4.9e-324 to 1.8e+308	double a,b;
	char	16 bits	true or false	char a;
	boolean	1 bit		true or false
27	What is Garbage collection? Objects are dynamically allocated by using the new operator, dynamically allocated objects must be manually released by use of a delete operator. Java takes a different approach; it handles deallocation automatically this is called garbage collection. When no references to an object exist, that object is assumed to be no longer needed, and the memory occupied by the object can be reclaimed. Garbage collection only occurs sporadically (if at all) during the execution of your program. It will not occur simply because one or more objects exist that are no longer used.			
28	What is difference between Methods and Constructor?(Nov 2017) A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator. A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.			
29	What is passed by reference?(May 2016) Objects are passed by reference. In java we can create an object pointing to a particular location ie NULL location by specifying : <class name> <object name>; and also can create object that allocates space for the variables declared in that particular class by specifying <i>Syntax:</i> <object name > = new <class name>();			
30	What is Constructors in Java? What are its types? A constructor is a special method that is used to initialize an object. The name of the constructor and the name of the class must be the same. A constructor does not have any return type. The constructor is invoked whenever an object of its associated class is created. It is called constructor because it creates the values for the data fields of the class. A constructor has same name as the class in which it resides. Constructor in Java cannot be abstract, static, final or synchronized. These modifiers are not allowed for constructors. <pre> Class Car { String name; String model; Car() //Constructor { name=""; model="";}} </pre> There are two types of Constructor Default Constructor Parameterized constructor Each time a new object is created at least one constructor will be invoked. <pre> Car c=new Car(); //Default constructor invoked Car c=new Car(name); //Parameterized constructor invoked </pre>			

31	<p>What is array? How to declare array and how to allocate the memory to for array?</p> <p>Java array contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a java array. Array in java is index based, first element of the array is stored at 0 index.</p> <p>data_type array_name []; and to allocate the memory- array_name=new data_type[size];where array_name represent name of the array, new is a keyword used to allocate the memory for arrays, data_type specifies the data type of array elements and size represents the size of an array. For example:int a=new int[10];</p>									
32	<p>Explain how to declare Two Dimensional array?</p> <p>The two dimensional arrays are the arrays in which elements are stored in rows as well as columns. For example:</p> <table><tr><td>10</td><td>20</td><td>30</td></tr><tr><td>40</td><td>50</td><td>60</td></tr><tr><td>70</td><td>80</td><td>90</td></tr></table> <p>The two dimensional array can be declared and initialized as follows <i>Syntax:</i> data_type array_name=new data_type[size];For example: int a[][]=new int[3][3];</p>	10	20	30	40	50	60	70	80	90
10	20	30								
40	50	60								
70	80	90								
33	<p>What is method in java? How to define and call the method?</p> <p>Method is a programming construct used for grouping the statement together to build a function. There are two ways by which the method is handled.</p> <p>1. Defining a method 2. Calling a method</p> <p>Here is example that helps to understand the concept of method defining and calling.</p> <pre>public class methDemo { public static void main(String args[]) { int a=10;int b=20;int c=sum(a,b); System.out.println("The sum of "+a+" and "+b+" is="+c); } public static int sum(int num1,int num2) { int ans=num1+num2; return ans;}}</pre>									
34	<p>What are public static void main(String args[]) and System.out.println() ?</p> <p>Public keyword is an access specifier. Static allows main() to be called without having to instantiate a particular instance of class. Void does not return any value. Main() is the method where java application begins.String args[] receives any command line arguments during runtime.System is a predefined class that provides access to the system. Out is output stream connected to console.println displays the output.</p>									
35	<p>What is down casting?</p> <p>Doing a cast from a base class to a more specific class. The cast does not convert the object, just asserts it actually is a more specific extended object.</p> <p>e.g. Dalamatian d = (Dalmatian) aDog;</p>									
36	<p>What are types of Constructors?</p> <p>Default Constructor, Parameterized Constructor, Copy Constructors</p>									
37	<p>What's the difference between an interface and an abstract class?</p> <p>An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.</p>									
38	<p>Explain about Static?</p>									

	<p>When a member is declared as static it can be accessed before any objects of its class are created and without any reference to any object. These are global variables, no copy of these variables can be made. static can also be declared for methods. and cannot refer to this or super.</p>
39	<p>List any four Java Doc comments. [NOV/DEC 2011]</p> <p>A Javadoc comment is set off from code by standard multi-line comment tags <code>/*</code> and <code>*/</code>. The opening tag, however, has an extra asterisk, as in <code>/**</code>. The first paragraph is a description of the method documented. Following the description are a varying number of descriptive tags, signifying: The parameters of the method (<code>@param</code>), What the method returns (<code>@return</code>) and any exceptions the method may throw (<code>@throws</code>)</p>
40	<p>What are the access specifiers/modifiers for classes in Java?</p> <p>Java Access Specifiers (also known as Visibility Specifiers) regulate access to classes, fields and methods in java. These specifiers determine whether a field or method in a class, can be used or invoked by another method in another class or sub-class. Access Specifiers can be used to restrict access. There are 4 types of java access modifiers: Private, Default, Protected and Public</p>
41	<p>What is a package? (May 2019)</p> <p>A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two forms, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.</p>
42	<p>What is static methods in Java?(May 2016)</p> <ul style="list-style-type: none"> • A static method belongs to the class rather than object of a class. • A static method can be invoked without the need for creating an instance of a class. • Static method can access static data member and can change the value of it. • There are two main restrictions for the static method is that the static method cannot use non-static data member or call non-static method directly.
43	<p>What are the control flow statements in java?</p> <p>A programming language uses control statements to control the flow of execution of a program based on certain conditions. These are used to cause the flow of execution to advance and branch based on changes to the state of a program.</p> <p>Java's Selection statements:</p> <ul style="list-style-type: none"> • if • if-else • nested-if • if-else-if • switch-case • jump – break, continue, return <p>These statements allow you to control the flow of your program's execution based upon conditions known only during run time.</p>

44	<p>What is static variables in Java?</p> <p>The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than instance of the class. If you declare any variable as static, it is known static variable.</p> <p>The static variable can be used to refer the common property of all objects (that is not unique foreach object) e.g. company name of employees, college name of students etc. The static variable gets memory only once in class area at the time of class loading.</p> <p>Advantage - It makes your program memory efficient (i.e it saves memory).</p>
----	--

UNIT-I / PART-B	
1	Differentiate between object oriented programming and procedure oriented programming(7) Write a java program to find the greatest of three number(6) (Dec 2022)
2	Explain constructor with an example(6) Write a java program to display the grade of the students by using get() method to get marks and compute() method to compute the average and display() method to display the grade of the student(7) (Dec 2022)
3	Explain the various features of the Object Oriented Programming Language(May 2019)
4	i) Describe the typical java program structure. ii) Explain the general java program compilation and execution.
5	What are the different data types in JAVA? Explain each of them with example.(May 2019)
6	Discuss in detail the access specifiers available in Java.(Nov 2016)
7	Explain Constructors with examples.(Nov 2017)
8	Explain in detail the various operators in Java.
9	Explain the concepts of arrays in Java and explain its types with examples?
10	Explain in detail about static variable and static method in Java with example? (Nov 2017)
11	Explain the control structures in java.(May 2019)

UNIT II INHERITANCE, PACKAGES AND INTERFACES Overloading Methods – Objects as Parameters – Returning Objects –Static, Nested and Inner Classes. Inheritance: Basics– Types of Inheritance -Super keyword -Method Overriding – Dynamic Method Dispatch –Abstract Classes – final with Inheritance. Packages and Interfaces: Packages – Packages and Member Access –Importing Packages – Interfaces.	
UNIT-II / PART-A	
1	Differentiate method overloading and overriding(Dec 2022) Method overloading is a type of static polymorphism. In Method overloading, we can define multiple methods with the same name but with different parameters. Method Overriding is a mechanism to achieve polymorphism where the super class and the sub-class have same methods, including the parameters and signature.
2	Can we access parent class variables in child class by using super keyword?(Dec 2022) Private methods of the super-class cannot be called. Only public and protected methods can be called by the super keyword. It is also used by class constructors to invoke constructors of its parent class. Super keyword are not used in static Method.
3	What is meant by Inheritance and what are its advantages? (May 2019) Inheritance is a relationship among classes, wherein one class shares the structure or behavior defined in another class. This is called Single Inheritance. If a class shares the structure or behavior from multiple classes, then it is called Multiple Inheritance. Inheritance defines “is-a” hierarchy among classes in which one subclass inherits from one or more generalized super classes. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.
4	What is the difference between superclass and subclass? A super class is a class that is inherited whereas sub class is a class that does the inheriting.
5	Differentiate between a Class and an Object? (Nov 2018) The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program. The Class class is used to obtain information about an object's design. A Class is only a definition or prototype of real life object. Whereas an object is an instance or living representation of real life object. Every object belongs to a class and every class contains one or more related objects.
6	Define super class and subclass? Super class is a class from which another class inherits. Subclass is a class that inherits from one or more classes.
7	What are the four types of access modifiers? (May 2018) There are 4 types of java access modifiers: <ol style="list-style-type: none"> 1. private 2. default 3. protected 4. public

8	<p>What is protected class in Java?</p> <p>A private member is only accessible within the same class as it is declared. A member with no access modifier is only accessible within classes in the same package. A protected member is accessible within all classes in the same package and within subclasses in other packages.</p>
9	<p>What is protected method?</p> <p>A protected method can be called by any subclass within its class, but not by unrelated classes. Declaring a method protected defines its access level. The other options for declaring visibility are private and public. If undeclared, the default access level is package.</p>
10	<p>What is final modifier? (May 2018)</p> <p>The final modifier keyword makes that the programmer cannot change the value anymore. The actual meaning depends on whether it is applied to a class, a variable, or a method.</p> <p>final Classes- A final class cannot have subclasses.</p> <p>final Variables- A final variable cannot be changed once it is initialized.</p> <p>final Methods- A final method cannot be overridden by subclasses.</p>
11	<p>What is a constructor in a class?</p> <p>In class-based object-oriented programming, a constructor is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.</p>
12	<p>Why creating an object of the sub class invokes also the constructor of the super class?</p> <p>When inheriting from another class, super() has to be called first in the constructor. If not, the compiler will insert that call. This is why super constructor is also invoked when a Sub object is created. This doesn't create two objects, only one Sub object. The reason to have super constructor called is that if super class could have private fields which need to be initialized by its constructor.</p>
13	<p>What is an Abstract Class?(Nov 2016) or When a class must be declared as abstract?(Dec 2022)</p> <p>A class which contains the abstract keyword in its declaration is known as abstract class. But, if a class has at least one abstract method, then the class must be declared abstract. If a class is declared abstract, it cannot be instantiated.</p>
14	<p>What are inner class and anonymous class?</p> <p>Inner class: classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private.</p> <p>Anonymous class: Anonymous class is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.</p>
15	<p>What is an Interface?</p> <p>Interface is an outside view of a class or object which emphasizes its abstraction while hiding its structure and secrets of its behavior.</p>
16	<p>What is a base class?</p> <p>Base class is the most generalized class in a class structure. Most applications have such root classes. In Java, Object is the base class for all classes.</p>

17	What is meant by Binding, Static binding, Dynamic binding? <i>Binding:</i> Binding denotes association of a name with a class. <i>Static binding:</i> Static binding is a binding in which the class association is made during compile time. This is also called as <i>Early binding</i> . <i>Dynamic binding:</i> Dynamic binding is a binding in which the class association is not made until the object is created at execution time. It is also called as <i>Late binding</i> .									
18	What is the difference between a static and a non-static inner class? A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.									
19	What is the difference between abstract class and interface?(Nov 2019) <table><tr><th>ABSTRACT CLASS</th><th>INTERFACE</th></tr><tr><td>1. Abstract class must have at least one abstract method and others may be concrete or abstract</td><td>All the methods declared inside an interface are abstract</td></tr><tr><td>2. In abstract class, key word abstract must be used for the methods</td><td>Interface we need not use that keyword for the methods.</td></tr><tr><td>3. Abstract class must have subclasses</td><td>Interface can't have subclasses</td></tr></table>		ABSTRACT CLASS	INTERFACE	1. Abstract class must have at least one abstract method and others may be concrete or abstract	All the methods declared inside an interface are abstract	2. In abstract class, key word abstract must be used for the methods	Interface we need not use that keyword for the methods.	3. Abstract class must have subclasses	Interface can't have subclasses
ABSTRACT CLASS	INTERFACE									
1. Abstract class must have at least one abstract method and others may be concrete or abstract	All the methods declared inside an interface are abstract									
2. In abstract class, key word abstract must be used for the methods	Interface we need not use that keyword for the methods.									
3. Abstract class must have subclasses	Interface can't have subclasses									
20	Can you have an inner class inside a method and what variables can you access? Yes, we can have an inner class inside a method and final variables can be accessed.									
21	What is interface and state its use? Interface is similar to a class which may contain method's signature only but not bodies and it is a formal set of method and constant declarations that must be defined by the class that implements it. Interfaces are useful for: 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.									
22	Difference between class and interface. <table><tr><th>CLASS</th><th>INTERFACE</th></tr><tr><td>1. Class are used to create new reference types</td><td>Interface are used to create new reference types</td></tr><tr><td>2. A class is a collection of fields and methods that operate on fields</td><td>An interface has fully abstract methods i.e. methods with nobody</td></tr><tr><td>3. Class can be instantiated</td><td>Interface can never be instantiated</td></tr></table>		CLASS	INTERFACE	1. Class are used to create new reference types	Interface are used to create new reference types	2. A class is a collection of fields and methods that operate on fields	An interface has fully abstract methods i.e. methods with nobody	3. Class can be instantiated	Interface can never be instantiated
CLASS	INTERFACE									
1. Class are used to create new reference types	Interface are used to create new reference types									
2. A class is a collection of fields and methods that operate on fields	An interface has fully abstract methods i.e. methods with nobody									
3. Class can be instantiated	Interface can never be instantiated									
23	What is extending interface? An interface can extend another interface in the same way that a class can extend another class. The extends keyword is used to extend an interface, and the child interface inherits the methods of the parent interface. Syntax: interface_name { Public void method1(): Public void method2(): } }									
24	What modifiers may be used with top-level class? Public, abstract and final can be used for top-level class.									

25	Define Package. (Nov 2019) To create a package is quite easy: simply include a package command as the first statement in a Java source file. Any classes declared within that file will belong to the specified package. The package statement defines a name space in which classes are stored. If you omit the package statement, the class names are put into the default package, which has no name.
26	How interfaces can be extended? One interface can inherit another by use of the keyword extends. The syntax is the same as for inheriting classes. When a class implements an interface that inherits another interface, it must provide implementations for all methods required by the interface inheritance chain.
27	Brief Inner class in Java with its syntax. Java inner class or nested class is a class which is declared inside the class or interface. We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable. Additionally, it can access all the members of outer class including private data members and methods. Syntax of Inner class <pre> class Java_Outer_class{ //code class Java_Inner_class{ //code } } </pre>
UNIT II / Part - B	
1	Write a java programs for library interface with drawbook() returnbook() and checkstatus() methods.(8) Explain Method overloading with an example(5) (Dec 2022)
2	What are the advantages of using packages?(5) Illustrate how to add classes in a packages and how to access these classes in another package.(8) (Dec 2022)
3	Explain the concept of inheritance with suitable examples.(May 2019)
4	State i) The properties of inheritance ii) The design hints for inheritance
5	Explain interfaces with example.(May 2019)
6	Differentiate method overloading and method overriding. Explain both with an example program.
7	Explain about the object and abstract classes with the syntax.
8	Explain the java program to demonstrate the Objects as Parameters – Returning Objects
9	Discuss in detail about inner class. With its advantages.(May 2019)
10	Explain how inner classes and anonymous classes works in java program. (May 2017)
11	What is a Package? What are the benefits of using packages? Write down the steps in creating a package and using it in a java program with an example.
12	Explain hierarchical and multi-level inheritance supported by java and demonstrate the execution order of constructors in these types.(May 2019,Dec 2019)
13	(I) Explain simple interfaces and nested interfaces with example (ii) Present a detailed comparison between classes and interfaces(May 2019, May 2018)

14	Explain about the object and abstract classes with the syntax.(Dec 2019)
15	Summarize the concept of supper classes and sub classes
16	Declare an abstract class to represent a bank account with data members name, account number, address and abstract methods withdraw and deposit. Method display() is needed to show balance. Derive a subclass Savings Account and add the following details: return on investment and the method calcAmt() to show the amount in the account after 1 year. Create instance of Savings Account and show the use of with draw and deposit abstract methods (Dec 2017)

UNIT-III/ PART-A

1	<p>Define Arithmetic Exception with example.(Dec 2022)</p> <p>ArithmeticException is an unchecked exception in Java. Usually, one would come across java. lang. ArithmeticException: / by zero which occurs when an attempt is made to divide two numbers and the number in the denominator is zero. ArithmeticException objects may be constructed by the JVM</p>										
2	<p>Name the two ways to create a thread in java. (Dec 2022)</p> <p>By extending the thread class By implementing a Runnable interface</p>										
3	<p>What are the types of errors?</p> <ul style="list-style-type: none"> • Compile time errors • Run time errors 										
4	<p>Define Java Exception.</p> <p>A Java exception is an object that describes an exceptional (that is, error) condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that exception is created and thrown in the method that caused the error.</p>										
5	<p>State the five keywords in exception handling.</p> <p>Java exception handling is managed via five keywords: try, catch, throw, throws, and finally.</p>										
6	<p>Draw the exception hierarchy.</p> <p>The top-level exception hierarchy is shown here:</p> <pre> graph BT Throwable[Throwable] --> Error[Error] Throwable --> Exception[Exception] Exception --> RuntimeException[RuntimeException] RuntimeException --> RedBox["NullPointerException NumberFormatException ClassCastException IndexOutOfBoundsException ..."] RuntimeException --> BlueBox["IOException SQLException MalformedURLException ..."] </pre>										
7	<p>Name any four java built in exceptions.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Exception</th><th style="text-align: center;">Meaning</th></tr> </thead> <tbody> <tr> <td>ArithmeticException</td><td>Arithmetic error, such as divide-by-zero.</td></tr> <tr> <td>ArrayIndexOutOfBoundsException</td><td>Arithmetic Exception Array index is out-of-bounds.</td></tr> <tr> <td>ArrayStoreException</td><td>Assignment to an array element of an incompatible type.</td></tr> <tr> <td>ClassCastException</td><td>Invalid cast</td></tr> </tbody> </table>	Exception	Meaning	ArithmeticException	Arithmetic error, such as divide-by-zero.	ArrayIndexOutOfBoundsException	Arithmetic Exception Array index is out-of-bounds.	ArrayStoreException	Assignment to an array element of an incompatible type.	ClassCastException	Invalid cast
Exception	Meaning										
ArithmeticException	Arithmetic error, such as divide-by-zero.										
ArrayIndexOutOfBoundsException	Arithmetic Exception Array index is out-of-bounds.										
ArrayStoreException	Assignment to an array element of an incompatible type.										
ClassCastException	Invalid cast										
8	<p>What is chained exception?</p> <p>Chained Exceptions allows to relate one exception with another exception, i.e one exception describes cause of another exception. For example, consider a situation in which a method throws an ArithmeticException because of an attempt to divide by zero but the actual cause of exception was an I/O error which caused the divisor to be zero.</p>										
9	<p>What does java.lang.StackTraceElement represent?</p> <p>The java.lang.StackTraceElement class element represents a single stack frame. All stack frames except for the one at the top of the stack represent a method invocation. The frame at the top of the stack represents the execution point at which the stack trace was generated.</p>										
10	<p>What are the useful methods of throwable classes</p> <ul style="list-style-type: none"> • public String getMessage() 										

	<ul style="list-style-type: none"> • public String getLocalizedMessage() • public synchronized Throwable getCause() • public String toString() • public void printStackTrace()
11	Compare throw and throws. <ul style="list-style-type: none"> • Throw is used to throw an exception & throws is used to declare an exception. • Throw is used in method implementation & throws is used in method signature. • Using throw keyword we can throw only 1 exception at a time & throws can declare multiple exceptions at a time.
12	What is the use of try and catch exception? Try-catch block is used for exception handling in the program code. try is the start of the block and catch is at the end of try block to handle the exceptions. A Program can have multiple catch blocks with a try and try-catch block can be nested also. catch block requires a parameter that should be of type Exception.
13	What is the use of finally exception? Finally block is optional and can be used only with try-catch block. Since exception halts the process of execution, we might have some resources open that will not get closed, so we can use finally block. finally block gets executed always, whether exception occurs or not.
14	How to write custom exception in Java? Extend Exception class or any of its subclasses to create our custom exception class. The custom exception class can have its own variables and methods and one can use to pass error codes or other exception related information to the exception handler.
15	What is OutOfMemoryError in Java? OutOfMemoryError in Java is a subclass of java.lang.VirtualMachineError and it's thrown by JVM when it ran out of heap memory.
16	What is difference between final, finally and finalize in Java? Final and finally are keywords in java whereas finalize is a method. Final keyword can be used with class variables so that they can't be reassigned, with class to avoid extending by classes and with methods to avoid overriding by subclasses. Finally keyword is used with try-catch block to provide statements that will always gets executed even if some exception arises, usually finally is used to close resources. finalize() method is executed by Garbage Collector before the object is destroyed, it's great way to make sure all the global resources are closed. Out of the three, only finally is related to java exception handling.
17	What happens when exception is thrown by main method? When exception is thrown by main() method, Java Runtime terminates the program and print the exception message and stack trace in system console.
18	Can we have an empty catch block? We can have an empty catch block but it's the example of worst programming. We should never have empty catch block because if the exception is caught by that block, we will have no information about the exception and it will be a nightmare to debug it.
19	How Java Exception Hierarchy categorized? Java Exceptions are hierarchical and inheritance is used to categorize different types of exceptions. Throwable is the parent class of Java Exceptions Hierarchy and it has two child objects – Error and Exception. Exceptions are further divided into checked exceptions and runtime exception.
20	How Threads are created in Java? Threads are created in two ways. They are by extending the Thread class and by implementing Runnable interface.

21	Define Thread? A thread is a single sequential flow of control within program. Sometimes, it is called an execution context or light weight process. A thread itself is not a program. A thread cannot run on its own. Rather, it runs within a program. A program can be divided into a number of packets of code, each representing a thread having its own separate flow of control.	
22	What is Multi-threading? Multithreading is a conceptual programming concept where a program(process) is divided into two or more subprograms(process), which can be implemented at the same time in parallel. A multithreaded program contains two or more parts that can run concurrently. Each part of such a program is called a thread, and each thread defines a separate path of execution.	
23	What is meant by Multitasking? Multitasking, in an operating system, is allowing a user to perform more than one computer task (such as the operation of an application program) at a time. The operating system is able to keep track of where you are in these tasks and go from one to the other without losing information. Multitasking is also known as multiprocessing.	
24	Difference between multi-threading and multi-tasking?	
	Multi-threading	Multi-tasking
	In any single process, multiple threads is allowed and again, can run simultaneously.	It refers to having multiple (programs, processes, tasks, threads) running at the same time.
	It is sharing of computing resources among threads of a single process.	It is sharing of computing resources (CPU, memory, devices, etc.) among processes
25	What do you mean by Thread Scheduling? Execution of multiple threads on a single CPU in some order is called scheduling. The Java runtime environment supports a very simple, deterministic scheduling algorithm called fixed- priority scheduling. This algorithm schedules threads on the basis of their priority relative to other Runnable threads.	
26	What is Synchronization thread? When two or more threads need access to a shared resource, they need some way to ensure that the resource will be used by only one thread at a time. The process by which this synchronization is achieved is called thread synchronization.	
27	What is thread priority? Every Java thread has a priority that helps the operating system determine the order in which threads are scheduled. Java priorities are in the range between MIN_PRIORITY (a constant of 1) and MAX_PRIORITY (a constant of 10). By default, every thread is given priority NORM_PRIORITY (a constant of 5) Threads with higher priority are more important to a program and should be allocated processor time before lower-priority threads. However, thread priorities cannot guarantee the order in which threads execute and very much platform independent.	
28	List out the methods of object class to perform inter thread communication? ➤ wait() – This method make the current thread to wait until another thread invokes the notify() method.	

	<ul style="list-style-type: none"> ➤ notify() – This method wakes up a thread that called wait() on same object. ➤ notifyAll() – This method wakes up all the thread that called wait() on same object. Wakes up all threads that are waiting on this object's monitor. <p>Above all three methods have been implemented as final method in Object class, so that they are available in all the classes in java world.</p>
29	<p>What are the various states of a thread? The following figure shows the states that a thread can be in during its life and illustrates which method calls cause a transition to another state.</p> <pre> graph TD Newborn[Newborn] -- start --> Running([Running]) Newborn -- stop --> Dead[Dead] subgraph Active_Thread [Active Thread] Running -- yield --> Runnable([Runnable]) end Running -- "suspend, sleep, wait" --> Blocked[Blocked] Blocked -- "resume, notify" --> Runnable Runnable -- stop --> Dead Blocked -- stop --> Dead subgraph Idle_Thread [Idle Thread (Not Runnable)] Blocked end subgraph Dead_State [Dead / Killed Thread] Dead end </pre>
30	<p>Why do we need run() and start() method both? Can we achieve it with only run method?</p> <ul style="list-style-type: none"> • The separate start() and run() methods in the Thread class provide two ways to create threaded programs. The start() method starts the execution of the new thread and calls the run() method. The start() method returns immediately and the new thread normally continues until the run() method returns. • The Thread class' run() method does nothing, so sub-classes should override the method with code to execute in the second thread. If a Thread is instantiated with a Runnable argument, the thread's run() method executes the run() method of the Runnable object in the new thread instead. • Depending on the nature of your threaded program, calling the Thread run() method directly can give the same output as calling via the start() method, but in the latter case the code is actually executed in a new thread.

Unit -III/Part B	
1	Create software for departmental stores to maintain the following details like item_no, item_description, requested quantity, cost price, Provide the options to update the stock. Calculate the selling price($SP = CP * 20\%$). Create an exception whenever the selling price of item exceeds the given amount (Dec 2022)
2	Discuss about try, catch and finally keywords in exception handling with an example (8) List and explain data types and their corresponding wrapper class(5)(Dec 2022)
3	Explain in detail the important methods of Java Exception Class?
4	Explain the different scenarios causing "Exception in thread main"?
5	How will you create your Own Exception Subclasses?
6	Explain in detail Chained exception with an example program.
7	Explain in detail the various exception types with its hierarchy.

8	Write programs to illustrate arithmetic exception, ArrayIndexOutOfBoundsException Exception and NumberFormat Exception.
9	Write a calculator program using exceptions and functions.
10	What are the two ways of thread creation? Explain with suitable examples.
11	With illustrations explain multithreading, interrupting threads, thread states and thread properties.
12	Describe the life cycle of thread and various thread methods.
13	Explain the thread properties in detail.
14	Explain inter thread communication and suspending, resuming and stopping threads.
15	Write a java program that synchronizes three different threads of the same program and displays the contents of the text supplies through the threads.
16	Write a java program for inventory problem to illustrate the usage of thread synchronized keyword and inter thread communication process. They have three classes called consumer, producer and stock.

Unit IV

Part A

1	<p>What is Thread Pool?(Dec 2022)</p> <p>A thread pool is a managed collection of threads that are available to perform tasks. Thread pools usually provide: Improved performance when executing large numbers of tasks due to reduced per-task invocation overhead. A means of bounding the resources, including threads, consumed when executing a collection of tasks.</p>
2	<p>What are input and output streams?</p> <p>An I/O Stream represents an input source or an output destination. A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs, and memory arrays.</p>
3	<p>What is a byte stream in java?</p> <p>Programs use byte streams to perform input and output of 8-bit bytes. All byte stream classes are descended from InputStream and OutputStream. There are many byte stream classes. The file I/O byte streams, are FileInputStream and FileOutputStream.</p>
4	<p>Define stream.</p> <p>A stream can be defined as a sequence of data. There are two kinds of Streams</p> <ul style="list-style-type: none">• InputStream – The InputStream is used to read data from a source.• OutputStream – The OutputStream is used for writing data to a destination.
5	<p>What is character stream?</p> <p>Character streams are used to perform input and output for 16-bit unicode. Though there are many classes related to character streams but the most frequently used classes are, FileReader and FileWriter.</p>
6	<p>What are directories in Java?</p> <p>A directory is a File which can contain a list of other files and directories. You use File object to create directories, to list down files available in a directory.</p>
7	<p>What are the two useful methods to create directories?</p> <p>There are two useful File utility methods, which can be used to create directories</p> <ul style="list-style-type: none">• The mkdir() method creates a directory, returning true on success and false on failure. Failure indicates that the path specified in the File object already exists, or that the directory cannot be created because the entire path does not exist yet.• The mkdirs() method creates both a directory and all the parents of the directory.
8	<p>State the use of java.io.Console.</p> <p>The java.io.Console class which provides convenient methods for reading input and writing output to the standard input (keyboard) and output streams (display) in command-line.</p>
9	<p>What is the use of java console class?</p> <p>The Java Console class is be used to get input from console. It provides methods to read texts and passwords. If you read password using Console class, it will not be displayed to the user. The java.io.Console class is attached with system console internally.</p>
10	<p>State the classes used to read file in java.</p> <p>The classes are:</p> <ul style="list-style-type: none">• FileReader for text files in your system's default encoding• FileInputStream for binary files and text files that contain 'weird' characters.

11	<p>What do you mean by generic programming?</p> <p>Generic programming is a style of computer programming in which algorithms are written in terms of to-be-specified-later types that are then instantiated when needed for specific types provided as parameters</p>
12	<p>Why do we need generics in Java?</p> <p>Code that uses generics has many benefits over non-generic code: Stronger type checks at compile time. A Java compiler applies strong type checking to generic code and issues errors if the code violates type safety. Fixing compile-time errors is easier than fixing runtime errors, which can be difficult to find.</p>
13	<p>State the two challenges of generic programming in virtual machines.</p> <ul style="list-style-type: none"> Generics are checked at compile-time for type-correctness. The generic type information is then removed in a process called type erasure. Type parameters cannot be determined at run-time
14	<p>When to use bounded type parameter?</p> <p>There may be times when you want to restrict the types that can be used as type arguments in a parameterized type. For example, a method that operates on numbers might only want to accept instances of Number or its subclasses. This is what bounded type parameters are for.</p>
15	<p>How to create generic class?</p> <p>A class that can refer to any type is known as generic class. Here, we are using T type parameter to create the generic class of specific type. Let's see the simple example to create and use the generic class.</p> <p>Creating generic class:</p> <pre>class MyGen<T>{ T obj; void add(T obj){ this.obj=obj;} T get(){return obj;} }</pre> <p>The T type indicates that it can refer to any type (like String, Integer, Employee etc.). The type you specify for the class, will be used to store and retrieve the data.</p>
16	<p>What is daemon thread?</p> <p>A daemon thread is a thread that does not prevent the JVM from exiting when the program finishes but the thread is still running. An example for a daemon thread is the garbage collection.</p>
17	<p>How to declare a java generic bounded type parameter?</p> <p>To declare a bounded type parameter, list the type parameter's name, followed by the extends keyword, followed by its upper bound, similar like below method.</p> <pre>Public static<T extends Comparable<T>> int compare(Tt1, Tt2) { return t1.compareTo(t2); }</pre> <p>The invocation of these methods is similar to unbounded method except that if we will try to use any class that is not Comparable, it will throw compile time error. Bounded type parameters can be used with methods as well as classes and interfaces</p>

18	What are wildcards in generics? In generic code, the question mark (?), called the wildcard, represents an unknown type. The wildcard can be used in a variety of situations: as the type of a parameter, field, or local variable; sometimes as a return type (though it is better programming practice to be more specific).																		
19	What is erasure in Java? Generics were introduced to the Java language to provide tighter type checks at compile time and to support generic programming. To implement generics, the Java compiler applies type erasure to : Replace all type parameters in generic types with their bounds or object if the types parameters are unbounded.																		
20	What are the restrictions on generics? To use Java generics effectively, you must consider the following restrictions: <ul style="list-style-type: none">• Cannot Instantiate Generic Types with Primitive Types• Cannot Create Instances of Type Parameters• Cannot Declare Static Fields Whose Types are Type Parameters• Cannot Use Casts or instance of With Parameterized Types• Cannot Create Arrays of Parameterized Types• Cannot Create, Catch, or Throw Objects of Parameterized Types• Cannot Overload a Method Where the Formal Parameter Types of Each Overload Erase to the Same Raw Type																		
21	Difference between String and StringBuffer. There are many differences between String and StringBuffer. A list of differences between String and StringBuffer are given below:																		
	<table><tr><th>No.</th><th>String</th><th>StringBuffer</th></tr><tr><td>1)</td><td>The String class is immutable.</td><td>The StringBuffer class is mutable.</td></tr><tr><td>2)</td><td>String is slow and consumes more memory when we concatenate too many strings because every time it creates new instance.</td><td>StringBuffer is fast and consumes less memory when we concatenate t strings.</td></tr><tr><td>3)</td><td>String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.</td><td>StringBuffer class doesn't override the equals() method of Object class.</td></tr><tr><td>4)</td><td>String class is slower while performing concatenation operation.</td><td>StringBuffer class is faster while performing concatenation operation.</td></tr><tr><td>5)</td><td>String class uses String constant pool.</td><td>StringBuffer uses Heap memory</td></tr></table>	No.	String	StringBuffer	1)	The String class is immutable.	The StringBuffer class is mutable.	2)	String is slow and consumes more memory when we concatenate too many strings because every time it creates new instance.	StringBuffer is fast and consumes less memory when we concatenate t strings.	3)	String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.	StringBuffer class doesn't override the equals() method of Object class.	4)	String class is slower while performing concatenation operation.	StringBuffer class is faster while performing concatenation operation.	5)	String class uses String constant pool.	StringBuffer uses Heap memory
	No.	String	StringBuffer																
	1)	The String class is immutable.	The StringBuffer class is mutable.																
	2)	String is slow and consumes more memory when we concatenate too many strings because every time it creates new instance.	StringBuffer is fast and consumes less memory when we concatenate t strings.																
	3)	String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.	StringBuffer class doesn't override the equals() method of Object class.																
	4)	String class is slower while performing concatenation operation.	StringBuffer class is faster while performing concatenation operation.																
5)	String class uses String constant pool.	StringBuffer uses Heap memory																	
22	Different between capacity and length method in string buffer The current length of a StringBuffer can be found via the length() method. The total allocated capacity can be found through the capacity() method. int capacity() Returns the current capacity. int length() Returns the length (character count).																		

Part B

1	Explain string buffer class with example(6) Outline parameter type bounds with an example (7) (Dec 2022)
2	Apply the string handling functions to do the following operations (i) Compare the two strings. (ii) Reverse the String."Happy"(Dec 2022)
3	Create a Central Library management system with libraries in different locations in a city. Each library can order a book from the central library and give it to the requesting customers. Availability of that book should be updated on each lending. Consider each library as a thread that requests for a book for purchase from the central library. If the requested book count is less than zero, then central library provides an appropriate message to the appropriate requesting library . Implement using Multithreading (Dec 2022)
4	Write a program to receive the name of a file within a text field and then displays its contents within a text area.
5	Write a Java program that prints the maximum of the sequence of non-negative integer values that are stored on the file data.txt.
6	Write a program reads every single character from the file MyFile.txt and prints all the characters to the output console also write an example program which uses a BufferedReader that wraps a FileReader to append text to an existing file.
7	Explain in detail about generic classes and methods in java with suitable example.
8	Describe briefly about generics with wildcards.
9	What are the restrictions are considered to use java generics effectively? Explain in detail.
10	Explain the all the string operators with syntax and example
11	Explain the string comparison methods with an example.
12	Explain the string method for searching and modifying the string.
13	Explain the string buffer methods with an example.

Unit V
Part A Questions with Answer

1. What is JavaFX?

JavaFX is a Java library used to build Rich Internet Applications. The applications written using this library can run consistently across multiple platforms. The applications developed using JavaFX can run on various devices such as Desktop Computers, Mobile Phones, TVs, Tablets, etc.

2. List out some of the important features of JavaFX.

- a. Written in Java
- b. FXML
- c. Scene Builder
- d. Swing Interoperability
- e. Built-in UI controls
- f. CSS like Styling
- g. Canvas and Printing API
- h. Rich set of API's
- i. Graphics pipeline

3. What do you mean by Scene Graph

In JavaFX, the GUI Applications were coded using a Scene Graph. A Scene Graph is the starting point of the construction of the GUI Application. It holds the (GUI) application primitives that are termed as nodes.

A node is a visual/graphical object and it may include –

- a. **Geometrical (Graphical) objects** – (2D and 3D) such as circle, rectangle, polygon, etc.
- b. **UI controls** – such as Button, Checkbox, Choice box, Text Area, etc.
- c. **Containers** – (layout panes) such as Border Pane, Grid Pane, Flow Pane, etc.
- d. **Media elements** – such as audio, video and image objects.

4. Define Prism

Prism is a **high performance hardware-accelerated graphical pipeline** that is used to render the graphics in JavaFX. It can render both 2-D and 3-D graphics.

To render graphics, a Prism uses –

- a. DirectX 9 on Windows XP and Vista.
- b. DirectX 11 on Windows 7.
- c. OpenGL on Mac and Linux, Embedded Systems.

5. List out the Lifecycle of JavaFX Application

The JavaFX Application class has three life cycle methods, which are –

- a. **start()** – The entry point method where the JavaFX graphics code is to be written.
- b. **stop()** – An empty method which can be overridden, here you can write the logic to stop the application.
- c. **init()** – An empty method which can be overridden, but you cannot create stage or scene in this method.

6. What are the three main aspects of user interface?

UI elements – These are the core visual elements which the user eventually sees and interacts with. JavaFX provides a huge list of widely used and common elements varying from basic to complex.

Layouts – They define how UI elements should be organized on the screen and provide a final look and feel to the GUI (Graphical User Interface).

Behavior – These are events which occur when the user interacts with UI elements.

7. list of commonly used controls while the GUI is designed using JavaFX.

- a. Label
- b. Button
- c. ColorPicker
- d. CheckBox
- e. RadioButton
- f. ListView
- g. TextField
- h. PasswordField
- i. Scrollbar
- j. FileChooser
- k. ProgressBar
- l. Slider

8. Write down the code to create a blank CheckBox

```
CheckBox checkbox = new CheckBox()
```

9. Write down the code to attach a label with the checkbox

```
CheckBox checkbox = new CheckBox("Label Name")
```

10. Which method used in the Text control to specify the position ?

We can set the position (origin) of the text by specifying the values to the properties x and y using their respective setter methods namely **setX()** and **setY()** like

```
text.setX(50);
```

```
text.setY(50);
```

11. What is the difference between radio button and ListView?

A radio button is displayed in a compact form that requires you to pull it down to see the list of available choices and only one item may be selected from a choice. A ListView may be displayed in such a way that several list items are visible and it supports the selection of one or more list items.

12. What do you mean by Layout.

After constructing all the required nodes in a scene, we will generally arrange them in order. This arrangement of the components within the container is called the Layout of the container. We can also say that we followed a layout as it includes placing all the components at a particular position within the container.

13. List out various Layout panes (classes) provided by JavaFX.

JavaFX provides several predefined layouts such as

HBox, VBox, Border Pane, Stack Pane, Text Flow, Anchor Pane, Title Pane, Grid Pane, Flow Panel, etc.

14. What are the properties of FlowPane?

- a. **alignment** – This property represents the alignment of the contents of the Flow pane. You can set this property using the setter method **setAlignment()**.

- b. **columnHalignment** – This property represents the horizontal alignments of nodes in a vertical flow pane.
 - c. **rowValignment** – This property represents the vertical alignment of nodes in a horizontal flow pane.
 - d. **Hgap** – This property is of double type and it represents the horizontal gap between the rows/columns of a flow pane.
 - e. **Orientation** – This property represents the orientation of a flow pane.
 - f. **Vgap** – This property is of double type and it represents the vertical gap between the rows/columns of a flow pane.
15. What are the properties of **BorderPane**?
- a. **bottom** – This property is of **Node** type and it represents the node placed at the bottom of the **BorderPane**. You can set value to this property using the setter method **setBottom()**.
 - b. **center** – This property is of **Node** type and it represents the node placed at the center of the **BorderPane**. You can set value to this property using the setter method **setCenter()**.
 - c. **left** – This property is of **Node** type and it represents the node placed at the left of the **BorderPane**. You can set value to this property using the setter method **setLeft()**.
 - d. **right** – This property is of **Node** type and it represents the node placed at the right of the **BorderPane**. You can set value to this property using the setter method **setRight()**.
 - e. **top** – This property is of **Node** type and it represents the node placed at the top of the **BorderPane**. You can set value to this property using the setter method **setTop()**.
 - f. What are the properties of **GridPane**?
 - g. **alignment** – This property represents the alignment of the pane and you can set value of this property using the **setAlignment()** method.
 - h. **hgap** – This property is of the type double and it represents the horizontal gap between columns.
 - i. **vgap** – This property is of the type double and it represents the vertical gap between rows.
 - j. **gridLinesVisible** – This property is of Boolean type. On true, the lines of the pane are set to be visible.

16. Define event

In JavaFX, we can develop GUI applications, web applications and graphical applications. In such applications, whenever a user interacts with the application (nodes), an event is said to have been occurred.

For example, clicking on a button, moving the mouse, entering a character through keyboard, selecting an item from list, scrolling the page are the activities that causes an event to happen.

17. What are the different classification of event?

The events can be broadly classified into the following two categories –

- a. **Foreground Events** – Those events which require the direct interaction of a user. They are generated as consequences of a person interacting with the graphical components in a Graphical User Interface. For example, clicking on a button,

moving the mouse, entering a character through keyboard, selecting an item from list, scrolling the page, etc.

- b. **Background Events** – Those events that don't require the interaction of end-user are known as background events. The operating system interruptions, hardware or software failure, timer expiry, operation completion are the example of background events.

18. What are the various types of events in JavaFX?

Mouse Event – This is an input event that occurs when a mouse is clicked. It includes actions like mouse clicked, mouse pressed, mouse released, mouse moved, mouse entered target, mouse exited target, etc.

Key Event – This is an input event that indicates the key stroke occurred on a node. This event includes actions like key pressed, key released and key typed.

Drag Event – This is an input event which occurs when the mouse is dragged. It includes actions like drag entered, drag dropped, drag entered target, drag exited target, drag over, etc.

Window Event – This is an event related to window showing/hiding actions. It includes actions like window hiding, window shown, window hidden, window showing, etc

19. What do mean by Event Handling ?

Event Handling is the mechanism that controls the event and decides what should happen, if an event occurs. This mechanism has the code which is known as an event handler that is executed when an event occurs

20. What are the method are available in Menu?

```
MenuBar menubar = new MenuBar(); //creating MenuBar
Menu MenuName = new Menu("Menu Name"); //creating Menu
MenuItem MenuItem1 = new MenuItem("Menu Item 1 Name"); //creating Menu
Item
MenuName.getItems().add(MenuItem1); //adding Menu Item to the Menu
Menubar.getMenus().add(MenuName); //adding Menu to the MenuBar
```

Part B

1. List and explain the important features of JavaFX
2. Explain the JavaFX Architecture in details.
3. Explain the JavaFX Application Structure in detail
4. Explain the following control with an example, **CheckBox**, **ToggleButton**, **RadioButtons**,
5. Explain the following control with an example, **ListView**, **ComboBox** , **ChoiceBox**, **Scroll panes**
6. Write a JavaFX program to create a registration form, which demonstrates controls in JavaFX such as **Date Picker**, **Radio Button**, **Toggle Button**, **Check Box**, **List View**, **Choice List**, etc.
7. Write a JavaFX program to demonstrates following layouts such as **HBox**, **VBox**, **Border Pane**, **Stack Pane**, **Text Flow**, **Anchor Pane**, **Title Pane**, **Grid Pane**, **Flow Panel**
8. Explain the Event Handling with example.
9. Write a Java program to create a menu bar and add menu to it and also add menuitems to the menu: