UNIT I INTRODUCTION TO OOP AND JAVA FUNDAMENTALS

Object Oriented Programming - Abstraction – objects and classes - Encapsulation- Inheritance - Polymorphism- OOP in Java – Characteristics of Java – The Java Environment - Java Source File - Structure – Compilation. Fundamental Programming Structures in Java – Defining classes in Java – constructors, methods -access specifiers - static members -Comments, Data Types, Variables, Operators, Control Flow, Arrays , Packages - JavaDoc comments.

UNIT-I/PART-A

1 **What are the OOP Principles?**

The principles of object-oriented programming is Class inheritance , interface implemen a ion, abstraction of data and behavior, encapsulation of data and class impl m a ion, polymorphism and virtual methods.

2 **What are the four cornerstones of OOP?**

Abstraction: Can manage complexity through abstraction. Gives the complete ov rvi w of a particular task and the details are handled by its derived classes. Example : Car

Encapsulation: 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

Inheritance: Is the process in which one object acquires the properties of a other object, ie., derived object. net

Polymorphism: One method different forms, ie., method overriding and interfaces are the examples of polymorphism.

3 **What are the features of Object Oriented Programming?**

- Emphasis is on data rather than procedure.
- Programs are divided into objects.
- Data structures are designed such that th y charact rize the objects.
- Functions that operate on the data of an obj ct are ti d tog ther.
- Data is hidden and cannot be acc ss d by xt rnal functions.
- Objects may communicate with ach oth r through functions.
- New data and functions can easily be a wh n v r necessary.
- Follows bottom-up approach.

4 **What are the features of J va L ngu ge?**

The features of Java L ngu ge re Simple, Object-Oriented, Portable, Platform independent, Secured, Robust, Architecture neutr l, Dyn mic, Interpreted, High Performance, Multithreaded and Distributed

5 **How Java supports latform inde endency?**

- The meaning of latform inde endent is that, the java source code can run on all operating systems a com iler is a rogram that translates the source code for another program from a programming language into executable code.
- This executable code may be 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 **Give the contents of Java Environment (JDK).**

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets.

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.

7 **What is Java Interpreter?**

	It is a Java Virtual Machine. An *interpreter* 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 {on the rocks} approach.
	C does not support overloading	JAVA supports Method Overloadi g
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 a d hence platform independent
	C++ supports structures, unions, templates, operator overloading, pointers and pointer arithmetic.	Java does not support pointers, templates, unions, o erator overloading, structures etc.
	C++supportdestructors,which is automatically invoked when the object is destroyed.	Java su ort automatic garbage collection. It do s not su ort destructors as C++ does.
10	**Distinguish between procedure ori nt 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 ccess specifier.	OOP has access specifiers named Public, Private, Protected, etc.
	POP does not have any ro er w y for hiding data so it is less secure	OOP provides Data Hiding so provides more security.
	In POP, Overloading is not ossible.	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 **Define Objects and Classes in Java**

- 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 he real world. The object can be declared by specifying the name of the class.

15 **Write the syntax for declaration of class and creation of objects?**

A class is declared using class keyword. A class contains both data and method that op rate o that data. Thus, the instance variables and methods are known as class members Wh cr ating an object from a class

Declaration − A variable declaration with a variable name with an object type

Instantiation − The 'new' keyword is used to create the object.

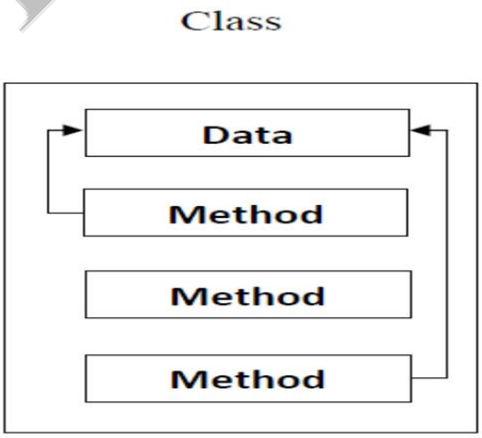
Initialization − The 'new' keyword is followed by a call to a constructor This call initializes the new object.

```
class Student
{
String name;
int rollno;
int age;
}
Student std=new Student();
```

- 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 dyn mic lly lloc tes memory for an object.

16 **Define Encapsulation (Apr/M y 2012) (Apr 2017)**

The wrapping up of data nd functions into 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.



17 **What is Inheritance? What are its types?**

- 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.

Types of inheritance are: Single inheritance, Multi-level inheritance, Hierarchical inheri ance, Hybrid inheritance.

Syntax :

```
class Subclass-name extends Superclass-name
{   //methods and fields }
```

19 **Define class[NOV/DEC 2011]**

Class is a template for a set of objects that share a common structure and a common behavio**r.**

20 **What do you mean by Dynamic Initialization?**

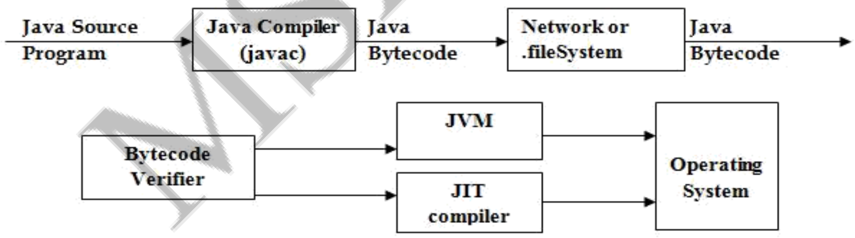
Java is a flexible programming language which allows the dynamic initialization of variables. In other words, at the time of declaration one can initiali e the variables. In java we can declare the variable at any place before it is used. Example: int a=10; float d=2.34f;

24 **What do you mean by Variable? What are the rul s for variable declaration?**

Variable is a fundamental unit of storage in java. The variabl s are used in combination with identifiers, data types, operators and some value for initialization.
The syntax of variable declaration will b : **data type name of variable[=initialization];**

21 **What are the steps for exection of a java program?**

A program is written in JAVA, the javac compil s 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 non-executable co e and needs an interpreter to execute on a machine. This interpreter is the JVM nd thus the Bytecode is executed by the JVM.
And finally program runs to give the esired output.



22 **What do you mean by Bytecode? What is JVM and JIT?**

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 hich 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.

23 **What is difference between Methods and Constructor?**

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.

24 **What are the different datatypes in java?**

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	int i,j;
long	64 bits	2,417,483,647	long x,y;
float	32 bits	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	float p,q;
double	64 bits	1.4e-045 to 3.4e+038	double a,b;
char	16 bits	4.9e-324 to 1.8e+308	char a;
boolean	1 bit	true or false	true or false

25 **What is Garbage collection?**

Objects are dynamically allocated by using the **new** operator, dynamically allocat d obj cts must be manually released by use of a **delete** operator. Java takes a differ t approach; it handles deallocation automatically this is called garbage collection When o r f re c s 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) duri g the execution of your program. It will not occur simply because one or more objects exist that are no longer used.

26 **What is difference between Methods and Constructor?**

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 op rator.

27 **What is passed by reference?**

Objects are passed by reference.In java we can cr ate an obj ct pointing to a particular location ie NULL location by specifying : <class nam > <obj ct nam >;and also can create object that allocates space for the variables eclared in that particular class by specifying

Syntax:<object name > = new <class nam >();

28 **What is Constructors in Java? What are its types?**

A constructor is a speci l method th t is used to initialize an object. The name of the constructor and the name of the class must be the s me. A constructor does not have any return type. The constructor invoked whenever n object of its associated class is created. It is called constructor because it creates the values for the ta fields of the class.

A constructor has same n me s the cl ss in which it resides. Constructor in Java cannot be abstract, static, final or synchronized. These modifiers are not allowed for constructors.

```
Class Car
{
String name;
String model;
Car()    //Constructor
{
name=””;
model=””;}}
```

There are t o types of Constructor

Default Constructor

Parameterized constructor

Each time a new object is created at least one constructor will be invoked.

Car c=new Car();   //Default constructor invoked

Car c=new Car(name); //Parameterized constructor invoked

29 **What is array? How to declare array and how to allocate the memory to for array?**

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.

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];

30 **Explain how to declare Two Dimensional array?**

The two dimensional arrays are the arrays in which elements are stored in rows as w ll as columns. For example:

10	20	30
40	50	60
70	80	90

padeepz

The two dimensional array can be declared and initialized as follows

Syntax: data_type array_name=new data_type[size];For exam le: int a[][]=new int[3][3];

31 **What is method in java? How to define and call the method?**

Method is a programming construct used for grou ing the statement together to build a function. There are two ways by which the method is handl d.

1. Defining a method 2. Calling method

Here is example that helps to understand the conc pt of m thod d fining and calling.

```
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;}}
```

32 **What are public static void main(String args[]) and System.out.println() ?**

Public key ord 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.

33 **What is do n casting?**

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.

e.g. Dalamatian d = (Dalmatian) aDog;

34 **What are types of Constructors?**

Default Constructor, Parameterized Constructor, Copy Constructors

35 **What's the difference between an interface and an abstract class?**

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.

36 **Explain about Static?**

	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**.
37	**List any four Java Doc comments. [NOV/DEC 2011]** A Javadoc comment is set off from code by standard multi-line comment tags /* and */. The opening tag, however, has an extra asterisk, as in /**.The first paragraph is a description of the method documented. Following the description are a varying number of descrip ive ags, signifying: The parameters of the method (@param),What the method returns (@return) and any exceptions the method may throw (@throws)
38	**What are the access specifiers/modifiers for classes in Java?** Java Access Specifiers (also known as Visibility Specifiers) regulate access to class s, fi lds and methods in java. These specifiers determine whether a field or method . inclass,can be us d 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 a d Public
39	**What is a package?** A java package is a group of similar types of classes, interfaces and sub-packages Package in java can be categorized in two form, built-in package and user-defined ackage There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.
40	**What is static methods in Java?** • A static method belongs to the class rather than obj ct of a class. • A static method can be invoked without the nfor cr ating an instance of a class. • Static method can access static data m mb r and can change the value of it. • There are two main restrictions for the static m thod is that the static method cannot use non static data member or call non-static m thod dir ctly.
41	**What are the control flow statements in java?** A programming language uses control st tements to control the flow of execution of program based on certain conditions. These re used to cause the flow of execution to advance and branch based on changes to the st te of program. Java's Selection statements: • if • if-else • nested-if • if-else-if • s itch-case • jump – break, continue, return These statements allow you to control the flow of your program's execution based upon conditions kno n only during run time.
42	**What is static variables in Java?** The static key ord in java is used for memory management mainly. We can apply java static key ord 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. The static variable can be used to refer the common property of all objects (that is not unique for each 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. Advantage - It makes your program memory efficient (i.e it saves memory).
	UNIT-I/PART-B
1	Explain the various features of the Object Oriented Programming Language

2	i) Describe the typical java program structure. ii) Explain the general java program compilation and execution.
3	What are the different data types in JAVA? Explain each of them with example.
4	How to pass and return the objects to and from the method?
5	Discuss in detail the access specifiers available in Java.
6	Explain Packages in detail.
7	Explain Constructors with examples.
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 exampl ?

UNIT II INHERITANCE AND INTERFACES

Inheritance – Super classes- sub classes –Protected members – constructors in sub class s- the Obj ct class – abstract classes and methods- final methods and classes – Interfaces – defi i g an i rfac , implementing interface, differences between classes and interfaces and exte di g i terfaces - Object cloning -inner classes, Array Lists - Strings

UNIT-II/PART-A	
1	**What is meant by Inheritance and what are its advantages?** Inheritance is a relationship among classes, wherein one class shares the structurenetorbehavior defined in another class. This is called Single Inheritance. If a class shares the structure or behavior from multiple classes, then it is called Multi le Inheritance. Inheritance defines "is-a" hierarchy among classes in which one subclass inh rits 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 subclass s.
2	**Brief about super ( ).** This is used to initialize constructor of base class from the derived class and also access the variables of base class like super.i = 10.
3	**What is the difference between superclass and subclass?** A super class is a class th t is inherited whereas sub class is a class that does the inheriting.
4	**Differentiate between Cl ss nd n Object?** The Object class is the highest-level cl ss in the Java class hierarchy. The Class class is used to represent the classes and interf ces th t re 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.
5	**Define super class and subclass?** Super class is a class from which another class inherits.Subclass is class that inherits from one or more classes.
6	**What are the four types of access modifiers?** There are 4 types of java access modifiers: 1. private 2. default 3. protected 4. public
7	**What is protected class in Java?** 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.
8	**What is protected function?** Protected members that are also declared as static are accessible to any friend or member function of a derived class. Protected members that are not declared as static are

	accessible to friends and member functions in a derived class only through a pointer to, reference to, or object of the derived class.
9	**What is protected method?** A protected method can be called by any subclass within its class, but not by unreleated 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.
10	**What is final modifier?** The final modifier keyword makes that the programmer cannot change the valuenetanymore.The actual meaning depends on whether it is applied to a class, a variable, or a method. final Classes- A final class cannot have subclasses. final Variables- A final variable cannot be changed once it is initialized. final Methods- A final method cannot be overridden by subclasses.
11	**What is a constructor in a class?** In class-based object-oriented programming, a constructor is a special type of subrouti called to create an object. Itpadeepzpreparesthenewobjectforuse,oftenaccepti g argume ts that the constructor uses to set required member variables.
12	**Why creating an object of the sub class invokes also the constructor of the super class?** 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 obj ct. The reason to have super constructor called is that if super class could have private fi lds which need to be initialized by its constructor.
13	**What is an Abstract Class?** Abstract class is a class that has no instanc s. An abstract class is written with the expectation that its concrete subclasses will a to its structure and b havior, typically by implementing its abstract operations.
14	**What are inner class and nonymous cl ss?** Inner class: classes defined in other cl sses, including those defined in methods are called inner classes. An inner class c n h ve ny ccessibility including private. Anonymous class: Anonymous class is a cl ss defined inside a method without a name and is instantiated and declared in the same lace and cannot have explicit constructors.
15	**What is an Interface?** Interface is an outside view of class or object which emphasizes its abstraction while hiding its structure and secrets of its behavior.
16	**What is a base class?** 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.
17	**What is meant by Binding, Static binding, Dynamic binding?** *Binding:* Binding denotes association of a name with a class. *Static binding:* Static binding is a binding in which the class association is made during compile time. This is also called as *Early* binding. *Dynamic binding*: 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 *Late* binding.
18	**What is reflection API? How are they implemented?** Reflection is the process of introspecting the features and state of a class at runtime and dynamically manipulate at run time. This is supported using Reflection API with built-in classes like Class, Method, Fields, Constructors etc. Example: Using Java Reflection API we can get the class name, by using the getName method.

19 **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.

20 **What is the difference between abstract class and interface?**

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

21 **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 access d

22 **What is interface and state its use?**

Interface is similar to a class which may contain method's signature only but ot bodies a d 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.

23 **Difference between class and interface.**

CLASS	INTERFACE
1. Class are used to create new reference types	Int rface are us d to create new r f r nce typ s
2. A class is a collection of fiel s and methods that operate on fields	An int rface has fully abstract methods i.e. methods with nobody
3. Class can be instanti ted	Interface can never be instantiated

24 **What is extending interf ce?**

An interface can extend nother interf ce in the same way that class can extend another class. The extends keyword is used to extend n interface, and the child interface inherits the methods of the parent interface

Syntax: interface interface_name{
Public void method1():
Public void method2(): }

25 **What modifiers may be used with top-level class?**

Public, abstract and final can be used for top-level class.

26 **What are the methods provided by the object class?**

The Object class provides five methods that are critical when writing multithreaded Java programs:

- notify
- notify All
- wait (three versions)

27 **What is a cloneable interface and how many methods does it contain?**

It is not having any method because it is a TAGGED or MARKER interface.

28 **Define: Dynamic proxy.**

A dynamic proxy is a class that implements a list of interfaces, which you specify at runtime when you create the proxy. To create a proxy, use the static method java.lang.reflect.Proxy::newProxyInstance().

This method takes three arguments:

	• The class loader to define the proxy class • An invocation handler to intercept and handle method calls • A list of interfaces that the proxy instance implements
29	**What is object cloning?** It is the process of duplicating an object so that two identical objects will exist in the memory at the same time.
30	**Define Package.** 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.
31	**How interfaces can be extended?** One interface can inherit another by use of the keyword extends. The sy tax is the same as for inheriting classes. When a class implements an interface that inherits another i terface, it must provide implementations for all methods required by the interface inherita ce chain
32	**Define Array list class.** The ArrayList class extends AbstractList and implements the List interface ArrayList is a generic class that has this declaration: class ArrayList<E> Here, E specifies the type of objects that the list will hold. An ArrayList is a variable-length array of obj ct r f r nc s. That is, an ArrayList can dynamically increase or decrease in size. Array lists are cr at d with an initial size. When this size is exceeded, the collection is automatically nlarg d. Wh n objects are removed, the array can be shrunk.

33	**Brief Inner class in Java with its syntax.**
	Java inner class or nested class is class which is clar d inside the class or interface.
	We use inner classes to logically group class s and int rfac s in one place so that it can be more readable and maintainable.
	Additionally, it can access ll the members of outer class including private data members and methods.
	Syntax of Inner class
	class Java_Outer cl ss{
	//code
	class Java_Inner class{
	//code
	}
	}

34	**What is String in Java? Is String is data type?** 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. 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.
	UNIT II / Part - B
1	Explain the concept of inheritance with suitable examples.
2	State i) The properties of inheritance ii) The design hints for inheritance
3	Explain interfaces with example.
4	Differentiate method overloading and method overriding. Explain both with an example program.
5	Differentiate method overloading and method overriding. Explain both with an example program.

6 Explain about the object and abstract classes with the syntax.

7 Discuss in detail about inner class. With its advantages.

8 What is meant by object cloning? Explain it with an example.

9 Explain how inner classes and anonymous classes works in java program.

10 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.

11 Explain arrays in java with suitable example.

12 How Strings are handled in java? Explain with code, the creation of Substring, Conca ena ion and testing for equality.

UNIT III EXCEPTION HANDLING AND I/O

Exceptions - exception hierarchy - throwing and catching exceptions – built-in exc ptions, cr a ing own exceptions, Stack Trace Elements. Input / Output Basics – Streams – Byte str ams and Character streams – Reading and Writing Console – Reading and Writing Files

UNIT-III/ PART-A

1 **What are the types of errors?**

- Compile time errors
- Run time errors

2 **Define Java Exception.**

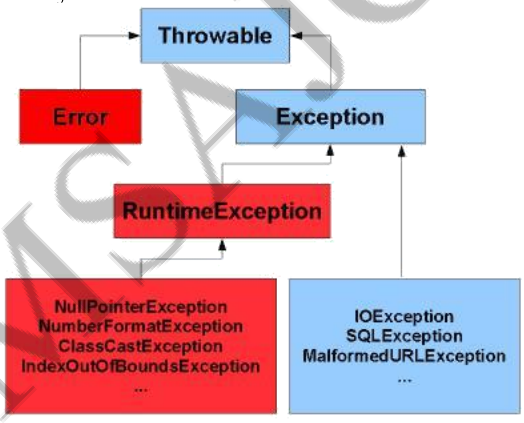
A Java exception is an object that describes an exce tional (that is, error) condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that exception is created and thrown in the method that caus the rror.

3 **State the five keywords in exception handling.**

Java exception handling is managed via five k ywords: try, catch, throw, throws, and finally.

4 **Draw the exception hierarchy.**

The top-level exception hierarchy is shown h r :



5 **Name any four java built in exceptions.**

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

6 **What is chained exception?**

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 thro s 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.

6 **What does java.lang.StackTraceElement represent?**

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.

7 **What are the useful methods of throwable classes**

- public String getMessage()

	- public String getLocalizedMessage() - public synchronized Throwable getCause() - public String toString() - public void printStackTrace()
8	**Compare throw and throws.** - 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.
9	**What is the use of try and catch exception?** Try-catch block is used for exception handling in the progam code. try is the start of he 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 param r that should be of type Exception.
10 15	**What is the use of finally exception?** Finally block is optional and can be used only with try-catch block Si ce exception halts the Can e have an emptypadeepzcatchblock? 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 occurrs or not
11	**How to write custom exception in Java?** Extend Exception class or any of its subclasses to create our custom exce tion 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 handl r.
12	**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.
13	**What is difference between final, finally and finalize in Java?** Final and finally are keywords in java wh r as finalize is method. **Final keyword** can be used with class variabl s so that they can't be reassigned, with class to avoid extending by classes and with metho s to avoid overriding by subclasses. **Finally keyword** is used with try-c tch block to provide statements that will always gets executed even if some exception rises, usually finally is used to close resources. **finalize()** method is executed by G rb ge Collector before the object is destroyed, it's great way to make sure all the glob l resources re closed. Out of the three, only finally is related to java exception handling
14	**What happens when exce tion is thrown by main method?** When exception is thro n by main() method, Java Runtime terminates the program and print the exception message and stack trace in system console.
	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.
16	**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.
17	**What are input and output streams?** 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.
18	**What is a byte stream in java?** 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.
19	**Define stream.** A stream can be defined as a sequence of data. There are two kinds of Streams • **InputStream** − The InputStream is used to read data from a source. • **OutputStream** − The OutputStream is used for writing data to a destination.
20	**What is character stream?** Character streams are used to perform input and output for 16-bit unicode. Though here are many classes related to character streams but the most frequently used classes are, FileReader and FileWriter.
21	**What are directories in Java?** A directory is a File which can contain a list of other files and directories. You use **File** obj ct to create directories, to list down files available in a directory.
22	**What are the two useful methods to create directories?** There are two useful **File** utility methods, which can be used to create directories • The **mkdir( )** method creates a directory, returning true on success a d 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 directory and all the arents of the directory.
23	**State the use of java.io.Console.** The **java.io.Console** class which provides convenient methods for reading input and writing output to the standard input (keyboard) and output str ams (dis lay) in command-line.
24	**What is the use of java console class?** 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 syst m console int rnally.
25	**State the classes used to read file in java.** The classes are: • FileReader for text files in your system's efault encoding • FileInputStream for bin ry files nd text files that contain 'weird' characters.
	Unit –III/Part B
1	Explain in detail the im ort nt methods of J va Exception Class?
2	Explain the different scen rios c using "Exception in thread main"?
3	How will you create your Own Exce tion Subclasses?
4	Explain in detail Chained exce tion with an example program.
5	Explain in detail the various exception types with its hierarchy.
6	Write programs to illustrate arithmetic exception, ArrayIndexOutOfBounds Exception and NumberFormat Exception.
7	Write a calculator program using exceptions and functions.
8	Create t o exception classes that can be used by the stack classes developed by TRY
9	Write a program to receive the name of a file within a text field and then displays its contents ithin a text area.
10	Write a Java program that prints the maximum of the sequence of non-negative integer values that are stored on the file data.txt.
11	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.

UNIT IV MULTITHREADING AND GENERIC PROGRAMMING

Difference between multi-threading and multi-tasking, thread life cycle, creating thread, synchronizing thread, inter-thread communication, demon thread, thread group, generic programming-generic classes-generic method-Bounded Types-Restrictions and Limitations

UNIT-IV / PART-A

1 **How Threads are created in Java?**

Threads are created in two ways. They are by extending the Thread class and by implementing Runnable interface.

2 **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 separa flow of control.

3 **What is Multi-threading?**

Multithreading is a conceptual programming concept where a program(process) is divid d into two or more subprograms(process), which can be implemented at the same time in parall l. A multithreaded program contains two or more parts that can run concurre tly Each part of such a program is called a thread, and each thread defines a separate path of execution

4 **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.

5 **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 r f rs to having multiple (programs, proc ss s, 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, m mory, devices, etc.) among processes

6 **What do you mean by Thread Sche uling?**

Execution of multiple thre ds on single CPU in some order is called scheduling. The Java runtime environment supports very simple, eterministic scheduling algorithm called fixed-priority scheduling. This lgorithm schedules threads on the basis of their priority relative to other Runnable threads.

7 **What is Thread Pool?**

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 overhea
- A means of bounding the resources, including threads, consumed when executing a

8 **What is Synchronization thread?**

When t o 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.

9 **What is thread priority?**

Every Java thread has a priority that helps the operating system determine the order in hich 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.

10 **List out the methods of object class to perform inter thread communication?**

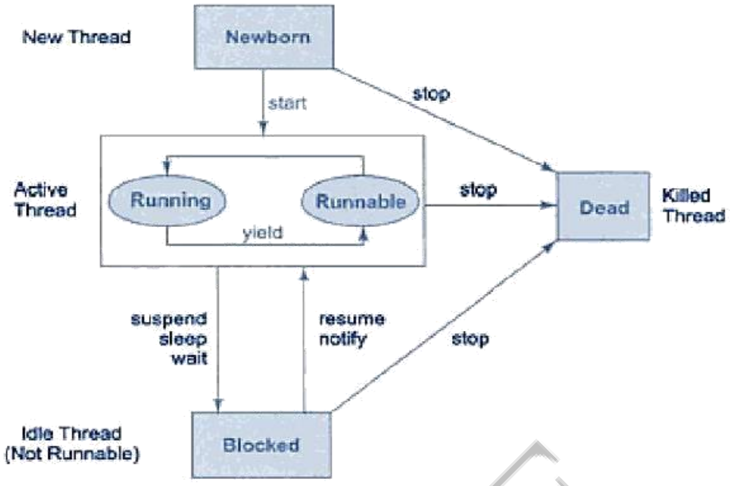
➢ **wait()** – This method make the current thread to wait until another thread invokes the notify() method.

- **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.

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.

11 **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.



12 **Why do we need run() and start() method both? Can we achi ve it with only run method?**

- The separate start() and run() methods in the Thr ad class rovide two ways to create threaded programs. The start() method starts the x cution of the new thread and calls the run() method. The start() method r turns imm diat ly and the new thread normally continues until the run() method returns.
- The Thread class' run() method o s nothing, so sub-classes should override the method with code to execute in the second thr ad. If a Thr ad 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 n ture of your threa ed program, calling the Thread run() method directly can give the s me out ut s c lling via the start() method, but in the latter case the code is actually executed in new thre d.

13 **What is daemon thread and which method is used to create the daemon thread?**

Daemon thread is a low riority thread which runs intermittently in the back ground doing the garbage collection o eration for the java runtime system. setDaemon method is used to create a daemon thread

14 **Write short note on isAlive() and join()?**

First, you can call isAlive() on the thread. This method is defined by Thread, and its general form is:

final Boolean isAlive()

The isAlive() method returns true if the thread upon which it is called is still running. It returns false other ise.

While **isAlive()** is occasionally useful, the method that you will more commonly use to ait for a thread to finish is called join(). The general form is:

final void join() throws InterruptedException

This method waits until the thread on which it is called terminates

15 **What do you mean by generic programming?**

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

16 **Define Deadlock and When it will occur?**

Deadlock describes a situation where two or more threads are blocked forever, waiting for each other. Deadlock occurs when multiple threads need the same locks but obtain them in different order. A Java multithreaded program may suffer from the deadlock condition because the **synchronized** keyword causes the executing thread to block while waiting for the lock, or monitor, associated with the specified object.

17 **Define thread group?**

Every Java thread is a member of a thread group. Thread groups provide a mechanism for collecting multiple threads into a single object and manipulating those threads all at once, rather than individually. For example, you can start or suspend all the threads wi hin a group with a single method call.

18 **Why do we need generics in Java?**

Code that uses **generics** has many benefits over non-generic code: Strong r type ch cks at compile time. A **Java** compiler applies strong type checking to generic code and issu s rrors if the code violates type safety. Fixing compile-time errors is easier than fixi g ru time rrors, which can be difficult to find.

19 **State the two challenges of generic programming in virtual machines**

- Generics are checked at compile-time for type-correctness. The ge eric type i formation is then removed in a process called type erasure.
- Type parameters cannot be determined at run-time

20 **When to use bounded type parameter?**

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 bound d ty arameters are for.

21 **How to create generic class?**

A class that can refer to any type is known as g n ric class. Here, we are using **T** type parameter to create the generic class of sp cific type.

Let's see the simple example to create and use the g n ric class.

Creating generic class:

```
class MyGen<T>{
T obj;
void add(T obj){this.obj=obj;}
T get(){return obj;} }
```

The T type indicates th t it c n refer to ny type (like String, Integer, Employee etc.). The type you specify for the class, will be used to store and retrieve the data.

22 **What is daemon thread?**

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.

23 **How to declare a java generic bounded type parameter?**

To declare a bounded type rameter, list the type parameter's name, followed by the extends key ord, follo ed by its upper bound, similar like below method.

```
Public static<T extends Comparable<T>>int compare(Tt1, Tt2){
        return t1.compareTo(t2);}
```

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

24 **What are wildcards in generics?**

In generic code, the question mark (?), called the wildcard, represents an unknown type. The ildcard 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).

25 **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 type parameters are unbounded.

25 **What are the restrictions on generics?**

To use Java generics effectively, you must consider the following restrictions:

- 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 Ov rload Erase o the Same Raw Type

Unit –IV/Part B

1 What are the two ways of thread creation? Explain with suitable examples

2 With illustrations explain multithreading, interrupting threads, thread states a d thread properties.

3 Describe the life cycle of thread and various thread methods.

4 Explain the thread properties in detail.

5 Explain inter thread communication and suspending, resuming and sto ing threads.

6 Write a java program that synchronizes three different threads of the same program and displays the contents of the text supplies through the thr ads.

7 Write a java program for inventory problem to illustrate the usage of thread synchronized keyword and inter thread communication proc ss. Th y have three classes called consumer, producer and stock.

8 Explain in detail about generic classes and m thods in java with suitable example.

9 Describe briefly about generics with wil car s.

10 What are the restrictions are consi ered to use java generics effectively? Explain in detail.

UNIT V EVENT DRIVEN PROGRAMMING

Graphics programming-Frame-Components-Working with 2D Shapes-Using color,fonts,and images-Basics of event handling-event h ndler- pter classes-actions-mouse events-AWT event hierarchy-introduction to Swing-layout m n gement-Swing components-Text fields, Text areas-Buttons-Check Boxes-Radio Buttons-Lists-choices-scrollbars-Windows-Menus-dialog Boxes.

Unit –V/Part A

1 **What is an Applet?**

Applet is a Java application, which can be executed in JVM, enabled web browsers.

2 **What are methods available in the Applet class?**

- init() - To initialize the applet each time it's loaded (or reloaded).
- start() - To start the applet's execution, such as when the applet's loaded or when the user revisits a page that contains the applet.
- stop() - To stop the applet's execution, such as when the user leaves the applet's page or quits the browser.
- paint()- To display the image
- destroy - To perform a final cleanup in preparation for unloading.

3 **Code a graphics method in java to draw the string “Hello World” from the Coordinates(100,200).**

g.dra String("Hello, World", 100, 150);

4 **What is AWT?**

A collection of graphical user interface (GUI) components that were implemented using native-platform versions of the components. These components provide that subset of functionality which is common to all native platforms. Largely supplanted by the Project Swing component set.

5 **What is the relationship between an event- listener interface and an event adapter class?**

- An event-listener interface allows describing the methods which must be implemented by one of the event handler for a specific event.
- An event-adapter allows default implementations of an event-listener interface of a specific event.

6 **List out some UI components available in AWT?**

- Buttons (java.awt.Button)
- Checkboxes (java.awt.Checkbox)
- Single-line text fields (java.awt.TextField)
- Larger text display and editing areas (java.awt.TextArea)
- Labels (java.awt.Label)
- Lists (java.awt.List)
- Pop-up lists of choices (java.awt.Choice)
- Sliders and scrollbars (java.awt.Scrollbar)
- Drawing areas (java.awt.Canvas)
- Menus (java.awt.Menu, java.awt.MenuItem, java.awt.CheckboxMe uItem)
- Containers (java.awt.Panel, java.awt.Window and its subclasses)

7 **Write some methods, which are used to add UI components in Applet?**

- add - Adds the specified Component.
- remove - Removes the specified Component.
- setLayout - Sets the layout manager.

8 **How can you prevent the overwriting of a displayed text in a TextField of a java program?**

If you create a TextField object with default text th n s tting the rom t text will not overwrite the default text.

To set the prompt text for a TextField use the s tPromptT xt m thod:

txtFld.setPromptText("Enter Name..");

To find out the value of the prompt text of a T xtFi ld obj ct use the getPromptText method:

String promptext = txtFld.getPromptT xt();

9 **How Events are handled in java ?**

A source generates n Event nd send it to one or more listeners registered with the source. Once event is received by the listener, they process the event and then return. Events are supported by a number of J va ck ges, like java.util, java.awt and java.awt.event.

10 **How does a radio button in j va differ from check box?**

- Radio buttons are used when there is a list of two or more options that are mutually exclusive and the user must select exactly one choice. In other words, clicking a non-selected radio button will deselect whatever other button was previously selected in the list.
- Checkboxes are used when there are lists of options and the user may select any number of choices, including zero, one, or several. In other words, each checkbox is independent of all other checkboxes in the list, so checking one box doesn't uncheck the others.

11 **Name the listener methods that must be implemented for the key listener interface. (NOV 2013)**

void keyTyped(KeyEvent e)
void keyPressed(KeyEvent e)
void keyReleased(KeyEvent e)

12 **Components of Event Handling**

Event handling has three main components,

Events : An event is a change in state of an object.

Events Source : Event source is an object that generates an event.

Listeners : A listener is an object that listens to the event. A listener gets notified when an event occurs.

13 **Mention the Differences between AWT and swing**

Java AWT	Java Swing
AWT components are platform-dependent.	Javaswingcomponentsare platform-independent.

	AWT components are heavyweight.	Swing components are lightweight.
	AWT provides less components than Swing.	Swing provides more powerful components such as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
	AWT doesn't follows MVC	Swing follows MVC.

14 **Mention some Event Classes and Interface**

Event Classes	Description	Listener Interface
ActionEvent	generated when button is pressed, menu-item is selected, list-item is double clicked	ActionListener net
MouseEvent	generated when mouse is dragged, moved,clicked,pressed or released and also when it enters or exit a component	MouseList r
KeyEvent	generated when input is received from keyboard	KeyListe r
ItemEvent	generated when check-box or list item is clicked	ItemListe er
TextEvent	generated when value of textarea or textfield is changed	TextListener
MouseWheelEvent	generated when mouse wheel is moved	MouseWheelListener
WindowEvent	generated when window is activated, deactivated, deiconified, iconified, o ened or closed	WindowListener
ComponentEvent	generated when compon nt is hidd n, moved, resized or set visible	ComponentEventListener
ContainerEvent	generated when compon nt is add or removed from contain r	ContainerListener

15 **What is a layout manager and what are iff r nt typ s of layout managers available in java AWT?**

A layout manager is an object that is used to organize components in a container. The different layouts are av il ble re FlowL yout, Bor erLayout, CardLayout, GridLayout and GridBagLayout.

16 **Define swing in java.**

Java Swing is a art of J va Found tion Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely.

17 **Why are swing com onents called lightweight component?**

Swing is considered lightweight because it is fully implemented in Java, without calling the native operating system for drawing the graphical user interface components.

18 **Mention some class for java swing**

JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser.

19 **What is the difference between scrollbar and scrollpane?**

A Scrollbar is a Component, but not a Container whereas Scrollpane is a Conatiner and handles its o n events and perform its own scrolling.

20 **What is the use of JButton and mention the constructor of JButton class.**

JButton class provides functionality of a button. JButton class has three constuctors,Button(Icon ic),JButton(String str),JButton(String str, Icon ic)

21 **What is the use of JTextField and mention the constructor of JTextField class.**

JTextField is used for taking input of single line of text. It is most widely used text component. It has three constructors, JTextField(int cols), JTextField(String str, int cols),JTextField(String str)
cols represent the number of columns in text field.

22 **What is meant by controls and what are different types of controls in AWT?**

Controls are components that allow a user to interact with your application and the AWT supports the following types of controls: Labels, Push Buttons, Check Boxes, Choice Lists, Lists, Scrollbars, Text Components. These controls are subclasses of Component.

<table>
<tr><td>23</td><td><b>What is the difference between choice and list?</b>
A Choice 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 List may be displayed in such a way that several list items are visible and it supports the selection of one or more list items.</td></tr>
<tr><td colspan="2"><b>UNIT-V / PART-B</b></td></tr>
<tr><td>1</td><td>What is event delegation model and what are the event classes and event interfaces?</td></tr>
<tr><td>2</td><td>Explain various components in AWT?</td></tr>
<tr><td>3</td><td>What is event handling in java? List out the available event classes and listener interfaces wi h suitable example.</td></tr>
<tr><td>4</td><td>Explain the layout managers in Java also describe the concept of menu creation.</td></tr>
<tr><td>5</td><td>What is an adapter class? Describe about various adapter classes in detail?</td></tr>
<tr><td>6</td><td>Explain about JComboBoxclass, JCheckBoxclass</td></tr>
<tr><td>7</td><td>Develop a java program that have 11 text fields one submit button. When you pr ss the button first 10 text field's average has to be displayed in the 11th text field.</td></tr>
<tr><td>8</td><td>Develop a java code that keeps the count of right clicks of mouse.</td></tr>
<tr><td>9</td><td>Explain about JButtonclass,padeepzJTextAreaclass,JFrameclass</td></tr>
<tr><td>10</td><td>Develop java program that changes the color of a filled circle when you make a right click.</td></tr>
<tr><td>11</td><td>An analysis of examination results at a school gave the following distribution of grades for all subjects taken in one year:
<table>
<tr><td colspan="2">GRADESPERCENTAGE</td></tr>
<tr><td>A10</td><td></td></tr>
<tr><td>B25</td><td></td></tr>
<tr><td>C45</td><td></td></tr>
<tr><td>D20</td><td></td></tr>
</table>
Write a java program to represent the distribution of ach grade in a pie chart , where each sliceof pie is differently colored.</td></tr>
<tr><td colspan="2">How will you display an image on the frame in window using java.12</td></tr>
</table>