DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited with 'A' Grade by NAAC, Accredited by TCS
Accredited by NBA with BME, ECE & EEE
PERAMBALUR - 621 212. Tamil Nadu.
website : www.dsengg.ac.in

DEPARTMENT OF INFORMATION
U20IT501 /JAVA PROGRAMMING
2-MARK Questions and Answers

UNIT I
CLASS, METHODS AND STRINGS

1) What is meant by Object Oriented Programming?

OOP is a method of programming in which programs are organised as cooperative collections of objects. Each object is an instance of a class and each class belong to a hierarchy.

2) What is a Class?

Class is a template for a set of objects that share a common structure and a common behaviour.

3) What is an Object?

Object is an instance of a class. It has state,behaviour and identity. It is also called as an instance of a class.

4) What is an Instance?

An instance has state, behaviour and identity. The structure and behaviour of similar classes are defined in their common class. An instance is also called as an object.

5) What are the core OOP's concepts?

Abstraction, Encapsulation,Inheritance and Polymorphism are the core OOP's concepts.

6) What is meant by abstraction?

Abstraction defines the essential characteristics of an object that distinguish it from all other kinds of objects. Abstraction provides crisply-defined conceptual boundaries relative to the perspective of the viewer. Its the process of focussing on the essential characteristics of an object. Abstraction is one of the fundamental elements of the object model.

7) What is meant by Encapsulation?

Encapsulation is the process of compartment alizing the elements of an abstraction that defines the structure and behavior. Encapsulation helps to separate the contractual interface of an abstraction and implementation.

8) What are Encapsulation, Inheritance and Polymorphism?

Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse. Inheritance is the process by which one object acquires the properties of another object. Polymorphism is the feature that allows one interface to be used for general class actions.

9) What are methods and how are they defined?

Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method. A method's signature is a combination of the first three parts mentioned above.

10) What are different types of access modifiers (Access specifiers)?

Access specifiers are keywords that determine the type of access to the member of a class. These keywords are for allowing

privileges to parts of a program such as functions and variables.
These are:
public: Any thing declared as public can be accessed from anywhere.
private: Any thing declared as private can't be seen outside of its class.
protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages.
default modifier : Can be accessed only to classes in the same package.

11) What is an Object and how do you allocate memory to it?

Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

12) Explain the usage of Java packages.

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

13) What is method overloading and method overriding?

Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding : When a method in a class having the same method name with same arguments is said to be method overriding.

14) What gives java it's "write once and run anywhere" nature?

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

15) What is a constructor? What is a destructor?

Constructor is an operation that creates an object and/or initialises its state. Destructor is an operation that frees the state of an object and/or destroys the object itself. In Java, there is no concept of destructors. Its taken care by the JVM.

16) What is the difference between constructor and method?

Constructor will be automatically invoked when an object is created whereas method has to be called explicitly

17) what are Static member classes?

A static member class is a static member of a class. Like any other static method, a static member class has access to all static methods of the parent, or top-level, class.

18) What is Garbage Collection and how to call it explicitly?

When an object is no longer referred to by any variable, java automatically reclaims memory used by that object. This is known as garbage collection. System. gc() method may be used to call it explicitly

19) In Java, How to make an object completely encapsulated?

All the instance variables should be declared as private and public getter and setter methods should be provided for accessing the instance variables.

20) What is static variable and static method?

static variable is a class variable which value remains constant for the entire class
static method is the one which can be called with the class itself and can hold only the staic variables

21) What is finalize() method?

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

22) What is the difference between String and String Buffer?

a) String objects are constants and immutable whereas StringBuffer objects are not. b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

23) What is the difference between Array and vector?

Array is a set of related data type and static whereas vector is a growable array of objects and dynamic

24) What is a package?

A package is a collection of classes and interfaces that provides a high-level layer of access protection and name space management.

25) What is the difference between this() and super()?

this() can be used to invoke a constructor of the same class .

super() can be used to invoke a super class constructor.

26) Explain working of Java Virtual Machine (JVM)?

JVM is an abstract computing machine like any other real computing machine which first converts .java file into .class file by using Compiler (.class is nothing but byte code file.) and Interpreter reads byte codes.

UNIT II
INHERITANCE, PACKAGE, INTERFACE AND EXCEPTION HANDLING

1) What is meant by Inheritance?

Inheritance is a relationship among classes, wherein one class shares the structure or behaviour defined in another class. This is called Single Inheritance. If a class shares the structure or behaviour 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 generalised superclasses.

2) What is meant by Inheritance and what are its advantages?

Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

3) 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.

4) Differentiate between a Class and an Object?

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.

5) What is meant by Binding?

Binding denotes association of a name with a class

6) What is meant by Polymorphism?

Polymorphism literally means taking more than one form. Polymorphism is a characteristic of being able to assign a different behavior or value in a subclass, to something that was declared in a parent class.

7) What is Dynamic Binding?

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding (also known as late binding) means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance.

8) What is final modifier?

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.

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.

9) What is an Abstract Class?

Abstract class is a class that has no instances. An abstract class is written with the expectation that its concrete subclasses will add to its structure and behaviour, typically by implementing its abstract operations.

10) What are inner class and anonymous class?

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. 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.

11) What is an Interface?

Interface is an outside view of a class or object which emphaizes its abstraction while hiding its structure and secrets of its behaviour.

12) What is a base class?

Base class is the most generalised class in a class structure. Most applications have such root classes. In Java, Object is the base class for all classes.

13) What is reflection in java?

Reflection allows Java code to discover information about the fields, methods and constructors of loaded classes and to dynamically invoke them.

14) Define superclass and subclass?

Superclass is a class from which another class inherits.

Subclass is a class that inherits from one or more classes.

15) 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.

16) 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.

17) 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.

18) What is the difference between abstract class and interface?

a) All the methods declared inside an interface are abstract whereas abstract class must have at least one abstract method and others may be concrete or abstract.

b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.

c) Abstract class must have subclasses whereas interface can't have subclasses.

19) 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.

20) What is interface and 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) 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.

21) How is polymorphism acheived in java?

Inheritance, Overloading and Overriding are used to acheive Polymorphism in java.

22) What modifiers may be used with top-level class?

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

23) 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.

24) What are the methods provided by the object class?

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

- notify
- notifyAll
- wait (three versions)

25) 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

26) 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.

UNIT III
GUI AND DATABASE CONNECTIVITY

1) What is the relationship between the Canvas class and the Graphics class?
A Canvas object provides access to a Graphics object via its paint() method.

2) How would you create a button with rounded edges?
There's 2 ways. The first thing is to know that a JButton's edges are drawn by a Border. so you can override the Button's paintComponent(Graphics) method and draw a circle or rounded rectangle (whatever), and turn off the border. Or you can create a custom border that draws a circle or rounded rectangle around any component and set the button's border to it.

3) What is the difference between the 'Font' and 'FontMetrics' class?
The Font Class is used to render 'glyphs' - the characters you see on the screen. FontMetrics encapsulates information about a specific font on a specific Graphics object. (width of the characters, ascent, descent)
4) What is the difference between the paint() and repaint() methods?
The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.

5) Which containers use a border Layout as their default layout?
The window, Frame and Dialog classes use a border layout as their default layout.

6) What is the difference between applications and applets?
a)Application must be run on local machine whereas applet needs no explicit installation on local machine.
b)Application must be run explicitly within a java-compatible virtual machine whereas applet loads and runs itself automatically in a java-enabled browser.
c)Application starts execution with its main method whereas applet starts execution with its init method.
d)Application can run with or without graphical user interface whereas applet must run within a graphical user interface.

7) Difference between Swing and Awt?
AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.

8) What is a layout manager and what are different types 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 available are FlowLayout, BorderLayout, CardLayout, GridLayout and GridBagLayout.

9) How are the elements of different layouts organized?
FlowLayout: The elements of a FlowLayout are organized in a top to bottom, left to right fashion.
BorderLayout: The elements of a BorderLayout are organized at the borders (North, South, East and West) and the center of a container.
CardLayout: The elements of a CardLayout are stacked, on top of the other, like a deck of cards.
GridLayout: The elements of a GridLayout are of equal size and are laid out using the square of a grid.
GridBagLayout: The elements of a GridBagLayout are organized according to a grid. However, the elements are of different size and may occupy more than one row or column of the grid. In addition, the

rows and columns may have different sizes. The *default* Layout Manager of Panel and Panel sub classes is FlowLayout.

10) Why would you use SwingUtilities.invokeAndWait or SwingUtilities.invokeLater?

I want to update a Swing component but I'm not in a callback. If I want the update to happen immediately (perhaps for a progress bar component) then I'd use invokeAndWait. If I don't care when the update occurs, I'd use invokeLater.

11) What is an event and what are the models available for event handling?

An event is an event object that describes a state of change in a source. In other words, event occurs when an action is generated, like pressing button, clicking mouse, selecting a list, etc. There are two types of models for handling events and they are: a) event-inheritance model and b) event-delegation model

12) 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 own events and perform its own scrolling.

13) Why won't the JVM terminate when I close all the application windows?

The AWT event dispatcher thread is not a daemon thread. You must explicitly call System.exit to terminate the JVM.

14) 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, and Text Components. These controls are subclasses of Component.

15) What is the difference between a Choice and a List?

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

16) What is the purpose of the enableEvents() method?

The enableEvents() method is used to enable an event for a particular object. Normally,an event is enabled when a listener is added to an object for a particular event. The enableEvents() method is used by objects that handle events by overriding their eventdispatch methods.

17) What is the difference between the File and RandomAccessFile classes?

The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

18) What is the lifecycle of an aplet?

init() method - Can be called when an applet is first loaded start() method - Can be called each time an applet is started. paint() method - Can be called when the applet is minimized or maximized. stop() method - Can be used when the browser moves off the applet's page. destroy() method - Can be called when the browser is finished with the applet.

19) What is the difference between a MenuItem and a CheckboxMenuItem?

The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.

20) What class is the top of the AWT event hierarchy?
The java.awt.AWTEvent class is the highest-level class in the AWT event-class hierarchy.

21) What is source and listener?
source : A source is an object that generates an event. This occurs when the internal state of that object changes in some way.
listener : A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

22) Explain how to render an HTML page using only Swing.
Use a JEditorPane or JTextPane and set it with an HTMLEditorKit, then load the text into the pane.

23) How would you detect a keypress in a JComboBox?
This is a trick. most people would say 'add a KeyListener to the JComboBox' - but the right answer is 'add a KeyListener to the JComboBox's editor component.'

24) What an I/O filter?
An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

25) How can I create my own GUI components?
Custom graphical components can be created by producing a class that inherits from java.awt.Canvas. Your component should override the paint method, just like an applet does, to provide the graphical features of the component.

UNIT IV
I/O AND THE COLLECTIONS FRAMEWORK

1) What is an exception?
An *exception* is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

2) What is error?
An Error indicates that a non-recoverable condition has occurred that should not be caught. Error, a subclass of Throwable, is intended for drastic problems, such as OutOfMemoryError, which would be reported by the JVM itself.

3) Which is superclass of Exception?
"Throwable", the parent class of all exception related classes.

4) What are the advantages of using exception handling?
Exception handling provides the following advantages over "traditional" error management techniques:
Separating Error Handling Code from "Regular" Code.
Propagating Errors Up the Call Stack.
Grouping Error Types and Error Differentiation.

5) What are the types of Exceptions in Java
There are two types of exceptions in Java, unchecked exceptions and checked exceptions.

Checked exceptions: A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses. Each method must either handle all checked exceptions by supplying a catch clause or list each unhandled checked exception as a thrown exception.
Unchecked exceptions: All Exceptions that extend the RuntimeException class are unchecked exceptions. Class Error and its subclasses also are unchecked.

6) Why Errors are Not Checked?

A unchecked exception classes which are the *error* classes (Error and its subclasses) are exempted from compile-time checking because they can occur at many points in the program and recovery from them is difficult or impossible. A program declaring such exceptions would be pointlessly.

7) How does a try statement determine which catch clause should be used to handle an exception?

When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

8) What is the purpose of the finally clause of a try-catch-finally statement?

The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

9) What is the difference between checked and Unchecked Exceptions in Java?

All predefined exceptions in Java are either a checked exception or an unchecked exception. Checked exceptions must be caught using try.. catch () block or we should throw the exception using throws clause. If you dont, compilation of program will fail.

10) What is the difference between exception and error?

The exception class defines mild error conditions that your program encounters. Exceptions can occur when trying to open the file, which does not exist, the network connection is disrupted, operands being manipulated are out of prescribed ranges, the class file you are interested in loading is missing. The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program terminate when such an error is encountered.

11) What is the catch or declare rule for method declarations?

If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

12) When is the finally clause of a try-catch-finally statement executed?

The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

13) What if there is a break or return statement in try block followed by finally Blocks?

If there is a return statement in the try block, the finally block executes right after the return statement encountered, and before the return executes.

14) What are the different ways to handle exceptions?

There are two ways to handle exceptions:

Wrapping the desired code in a try block followed by a catch block to catch the exceptions.

List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions.

15) How to create custom exceptions?
By extending the Exception class or one of its subclasses.
Example:

```
class MyException extends Exception {
  public MyException() { super(); }
  public MyException(String s) { super(s); }
  }
```

16) Can we have the try block without catch block?
Yes, we can have the try block without catch block, but finally block should follow the try block.
Note: It is not valid to use a try clause without either a catch clause or a finally clause.

17) What is the difference between swing and applet?
Swing is a light weight component whereas Applet is a heavy weight Component. Applet does not require main method, instead it needs init method.

18) What is the use of assert keyword?
Assert keyword validates certain expressions. It replaces the if block effectively and throws an AssertionError on failure. The assert keyword should be used only for critical arguments (means without that the method does nothing).

19) How does finally block differ from finalize() method?
Finally block will be executed whether or not an exception is thrown. So it is used to free resoources. finalize() is a protected method in the Object class which is called by the JVM just before an object is garbage collected.

20) What is the difference between throw and throws clause?
throw is used to throw an exception manually, where as throws is used in the case of checked exceptions, to tell the compiler that we haven't handled the exception, so that the exception will be handled by the calling function.

21) What are the different ways to generate and Exception?
There are two different ways to generate an Exception.
Exceptions can be generated by the Java run-time system.
Exceptions thrown by Java relate to fundamental errors that violate the rules of the Java language or the constraints of the Java execution environment.
Exceptions can be manually generated by your code.
Manually generated exceptions are typically used to report some error condition to the caller of a method.

22) Where does Exception stand in the Java tree hierarchy?
java.lang.Object
java.lang.Throwable
java.lang.Exception
java.lang.Error

23) WhatStackOverflowError?

The StackOverFlowError is an Error Object thorwn by the Runtime System when it Encounters that your application/code has ran out of the memory. It may occur in case of recursive methods or a large amount of data is fetched from the server and stored in some object. This error is generated by JVM.

e.g. void swap(){
swap();
}

24) Explain the exception hierarchy in java.

The hierarchy is as follows: Throwable is a parent class off all Exception classes. They are two types of Exceptions: Checked exceptions and UncheckedExceptions. Both type of exceptions extends Exception class

25) How do you get the descriptive information about the Exception occurred during the program execution?

All the exceptions inherit a method printStackTrace() from the Throwable class. This method prints the stack trace from where the exception occurred. It prints the most recently entered method first and continues down, printing the name of each method as it works its way down the call stack from the top.

UNIT V
THREADS, GENERICS AND FUNCTIONAL PROGRAMMING

1) Explain different way of using thread?

The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance..the only interface can help.

2) What are the different states of a thread ?

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

3) Why are there separate wait and sleep methods?

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

4) What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?

Multithreading is the mechanism in which more than one thread run independent of each other within the process. wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class. wait() : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state. notify() or notifyAll() : To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.

5) What is synchronization and why is it important?

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

6) How does multithreading take place on a computer with a single CPU?

The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

7) What is the difference between process and thread?

Process is a program in execution whereas thread is a separate path of execution in a program.

8) What happens when you invoke a thread's interrupt method while it is sleeping or waiting?

When a task's interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.

9) How can we create a thread?

A thread can be created by extending Thread class or by implementing Runnable interface. Then we need to override the method public void run().

10) What are three ways in which a thread can enter the waiting state?

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

11) How can i tell what state a thread is in ?

Prior to Java 5, isAlive() was commonly used to test a threads state. If isAlive() returned false the thread was either new or terminated but there was simply no way to differentiate between the two.

12) What is synchronized keyword? In what situations you will Use it?

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

13) What is serialization?

Serialization is the process of writing complete state of java object into output stream, that stream can be file or byte array or stream associated with TCP/IP socket.

14) What does the Serializable interface do?

Serializable is a tagging interface; it prescribes no methods. It serves to assign the Serializable data type to the tagged class and to identify the class as one which the developer has designed for persistence. ObjectOutputStream serializes only those objects which implement this interface.

15) When you will synchronize a piece of your code?

When you expect your code will be accessed by different threads and these threads may change a particular data causing data corruption.

16) What is daemon thread and which method is used to create the daemon thread?

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

17) What is the difference between yielding and sleeping?

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

18) What is casting?
There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

19) What classes of exceptions may be thrown by a throw statement?
A throw statement may throw any expression that may be assigned to the Throwable type.

20) A Thread is runnable, how does that work?
The Thread class' run method normally invokes the run method of the Runnable type it is passed in its constructor. However, it is possible to override the thread's run method with your own.

21) Can I implement my own start() method?
The Thread start() method is not marked final, but should not be overridden. This method contains the code that creates a new executable thread and is very specialised. Your threaded application should either pass a Runnable type to a new Thread, or extend Thread and override the run() method.

22) Do I need to use synchronized on setValue(int)?
It depends whether the method affects method local variables, class static or instance variables. If only method local variables are changed, the value is said to be *confined* by the method and is not prone to threading issues.

23) What is thread priority?
Thread Priority is an integer value that identifies the relative order in which it should be executed with respect to others. The thread priority values ranging from 1- 10 and the default value is 5. But if a thread have higher priority doesn't means that it will execute first. The thread scheduling depends on the OS.

24) What are the different ways in which a thread can enter into waiting state?
There are three ways for a thread to enter into waiting state. By invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method.

25) How would you implement a thread pool?
The ThreadPool class is a generic implementation of a thread pool, which takes the following input Size of the pool to be constructed and name of the class which implements Runnable (which has a visible default constructor) and constructs a thread pool with active threads that are waiting for activation. once the threads have finished processing they come back and wait once again in the pool.

26) What is a thread group?
A thread group is a data structure that controls the state of collection of thread as a whole managed by the particular runtime environment.

U20IT501 /JAVA PROGRAMMING

QUESTION BANK
PART-A & PART-B IMPORTANT QUESTIONS

UNIT I
PART A

1. What are the OOP Principles?
2. What is Encapsulation?
3. What is Polymorphism?
4. What is Inheritance?
5. What are the features of Java Language?
6. What is the need for Java Language?
7. What is platform independency?
8. What is Architecture Neutral?
9. How Java supports platform independency?
10. Why Java is important to Internet?
11. What are the types of programs Java can handle?
12. What is an applet program?
13. Compare Application and Applet.
14. What are the advantages of Java Language?
15. Give the contents of Java Environment (JDK).
16. Give any 4 differences between C and Java.
17. Give any 4 differences between C++ and Java.
18. What are the different types of comment symbols in Java?
19. What are the data types supported in Java?
20. What is the difference between a char in C/C++ and char in Java?
21. How is a constant defined in Java?
22. What is the use of final keyword?
23. What are the different types of operators used in Java?
24. What is short-Circuit operator?
25. What is labeled break?
26. What is the use of for each control structure?
27. What is the need for static variables?
28. What is the need for static methods?
29. Compare static constants and final constants.
30. Why is main method assigned as public?
31. Why is main method assigned as static?
32. What are the types of variables Java handles?
33. What are the relationships between classes?
34. What is the general form of a class?
35. What is the use of new keyword?
36. If ObjA1 is an object of class A created using new keyword, What does the statement A ObjA2=ObjA1; mean?
37. What is a constructor?
38. What is the difference between a constructor and a method?
39. What is the use of this keyword?
40. What are destructors?
41. How is object destruction done in Java?
42. What is Garbage collection?
43. What is the use of finalize method?
44. Compare Garbage collection and finalize method?
45. How is it guaranteed that finalize methods are called?
46. What is method overloading?
47. What is a String in Java?
48. What is the difference between a String in Java and String in C/C++?
49. Name a few String methods.
50. What is the difference between Concat method and + operator to join strings?

51. What is String Buffer?
52. How does String class differ from the String Buffer class?
53. Name some methods available under String Buffer class.
54. Output of some expressions using String methods.
55. How will you initialize arrays?
56. What is arraycopy method? Explain with syntax.
57. What are the methods under Util.Arrays?
58. Use the array sort method to sort the given array.
59. Give the syntax for array fill operation.
60. What is vector? How is it different from an array?
61. What is the constraint for using vectors?
62. What is wrapper class?
63. What are the different access specifiers available in Java?
64. What is the default access specifier in Java?
65. What is a package in Java?
66. Name some Java API Packages.
67. Name some JavaDoc Comments.
68. What is CommandLine Arguments.

Part B

1. Explain OOP Principles.
2. Explain the features of Java Language.
3. Compare and Contrast Java with C++.
4. Explain Constructors with examples.
5. Explain the methods available under String and String Buffer Class.
6. Explain the Date Class methods with examples.
7. Discuss in detail the access specifiers available in Java.
8. Explain the different visibility controls and also compare with each of them.
9. Explain the different methods in java.Util.Arrays class with example.
10. Explain Packages in detail.
11. Discuss the methods under Array Class.
12. Discuss some of the classes available under Lang package.
13. Illustrate with examples: static and final.
14. Explain method overriding with example program.
15. What is javaDoc? Explain the comments for classes, methods, fields and link.
16. Application Programs in Java.

UNIT II

PART A

1. Define Inheritance
2. What are the types of inheritance?
3. How is multiple inheritance achieved in java?
4. What is the use of super keyword?
5. What are object wrappers? Give example.
6. What is Inheritance Hierarchy?
7. Differentiate overloading and overriding.
8. Define polymorphism.
9. Differentiate static binding and dynamic binding.
10. When will a class be declared as final?
11. When will a method be declared final?
12. What is an abstract class?
13. What is the need for abstract classes?
14. Explain about protected visibility control.
15. What are the methods under "object" class / java.lang.Object.
16. Explain to String method of object class.
17. What is reflection?
18. What are the uses of reflection in Java.

19. How will you create an instance of Class.
20. What are the methods under reflection used to analyze the capabilities of classes?
21. How to create arrays dynamically using reflection package.
22. Define an interface.
23. What is the need for an interface?
24. What are the properties of an interface?
25. Differentiate Abstract classes and interface.
26. What is object cloning?
27. Differentiate cloning and copying.
28. Differentiate shallow copy and deep copy in cloning.
29. Does Inheritance removes any fields/or methods of super class?
30. Mention the use of final keyword.
31. What is nested class? Mention its types.
32. What is inner class?
33. What is the need for inner classes?
34. What are the rules for inner class?
35. What is local inner class and anonymous inner class? Give their advantages.
36. Write the advantages and disadvantages of static nested class.
37. Define proxies.
38. Write the application of proxies.
39. What are the properties of proxy classes?

PART B

1. Explain the concept of inheritance and its types.
2. Explain the concept of overriding with examples.
3. What is dynamic binding? Explain with example.
4. Explain the uses of reflection with examples.
5. Define an interface. Explain with example.
6. Explain the methods under “object” class and “class” class.
7. What is object cloning? Explain deep copy and shallow copy with examples.
8. Explain static nested class and inner class with examples.
9. With an example explain proxies.
10. Develop a message abstract class which contains playMessage abstract method. Write a different sub-classes like TextMessage, VoiceMessage and FaxMessage classes for to implementing the playMessage method.
11. Develop a abstract Reservation class which has Reserve abstract method. Implement the sub-classes like ReserveTrain and ReserveBus classes and implement the same.
12. Develop an Interest interface which contains simpleInterest and compInterest methods and static final field of Rate 25%. Write a class to implement those methods.
13. Develop a Library interface which has drawbook(), returnbook() (with fine), checkstatus() and reservebook() methods. All the methods tagged with public.
14. Develop an Employee class which implements the Comparable and Cloneable interfaces. Implement the sorting of persons (based on name in alphabetical). Also implement the shallow copy (for name and age) and deep copy (for DateOfJoining).
15. Explain the different methods supported in Object class with example.
16. Explain the methods supported in Class class.
17. Explain the Methods supported in reflect package. Also write a program to implement the reflection of a particular class details like constructors, methods and fields with its modifiers.
18. Develop a static Inner class called Pair which has MinMax method for finding min and max values from the array.
19. What is proxy class? Develop a code for constructing a proxy objects to trace a binary search method with explanations.

UNIT III
PART A

1. Draw the inheritance hierarchy for the frame and component classes in AWT and Swing.
2. What are the advantages of using swing over awt?
3. How do achieve special fonts for your text? Give example.
4. Give the syntax of drawImage() and copyArea() methods.
5. What is Adapter class?
6. Draw the AWT event Hierarchy.
7. What are the swing components?
8. What are the methods under Action Interface.
9. What are the methods under WindowListener Interface.
10. What is the difference between Swing and AWT?

PART B

1. Explain the classes under 2D shapes.
2. Explain event handling with examples.
3. Explain action event with an example.
4. What are the swing components. Explain.
5. Describe the AWT event hierarchy.

UNIT IV
PART A

1. What is generic programming?
2. What are Checked and UnChecked Exception?
3. What are checked exceptions?
4. What are runtime exceptions?
5. What is the difference between error and an exception?
6. What classes of exceptions may be caught by a catch clause?.
7. If I want an object of my class to be thrown as an exception object, what should I do?
8. How to create custom exceptions?
9. What are the different ways to handle exceptions?
10. What is the purpose of the finally clause of a try-catch-finally statement?
11. What is the basic difference between the 2 approaches to exception handling.
12. Is it necessary that each try block must be followed by a catch block?
13. How does Java handle integer overflows and underflows?

PART B

1. Explain generic classes and methods.
2. Explain exception hierarchy.
3. What are the advantages of Generic Programming?
4. Explain the different ways to handle exceptions.
5. How Java handle overflows and underflows?

UNIT V
PART A

1. Describe synchronization in respect to multithreading.
2. Explain different way of using thread?
3. What is synchronization and why is it important?
4. When a thread is created and started, what is its initial state?
5. What are synchronized methods and synchronized statements?
6. What is daemon thread and which method is used to create the daemon thread?
7. What method must be implemented by all threads?
8. What kind of thread is the Garbage collector thread?
9. What is a daemon thread?
10. What is a thread?
11. What is the algorithm used in Thread scheduling?
12. What are the different level lockings using the synchronization keyword?
13. What are the ways in which you can instantiate a thread?
14. What are the states of a thread?
15. What are the threads will start, when you start the java program?

16. What are the different identifier states of a Thread?
17. Why do threads block on I/O?
18. What is synchronization and why is it important?

PART B

1. Explain the different states of a thread.
2. Explain thread synchronization with examples.
3. Explain the algorithm used for thread scheduling.
4. Describe multi threading.
5. Explain Deadlocks.