UNIT I

PART A

1. **Define classes in java**

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

2. **Name the types of Array**

- Single Dimensional Array
- Two Dimensional Array
- Three Dimensional Array
- Character array

3. **List any four java Doc comments**

Javadoc is a tool which comes with JDK and it is used for generating Java code documentation in

HTML format from Java source code, which requires documentation in a predefined format. Following is a simple example where the lines inside /*….*/ are Java multi-line comments. Similarly, the which preceeds // is Java single-line comment.

4. **Define access specifier? give example**

These Specifiers determine whether a field or method in a class, can be used or invoked by another method in another class or sub-class

Types Of Access Specifiers :

In java we have four Access Specifiers and they are listed below.

1. public
2. private
3. protected
4. default(no specifier)

5. **Define objects and object variable**

The Object is the instance itself, whereas the Object Variable is the reference to the Object. Here's a contrived example: Object o = new Object(); Object ref1 = o; In his case, there is a single instance of the Object, but it is referenced by two Object Variables: o and ref1.

6. **What is the need of overloaded constructors**

The purpose of constructor is to initialize the object of a class while the purpose of a method is to perform a task by executing java code. Constructors cannot be abstract, final, static and synchronised while methods can be. Constructors do not have return types while methods do.

7. **Describe default constructor**

A *default constructor* is a *constructor* that either has no parameters, or if it has parameters, all the parameters have *default* values. ... This *constructor* is an inline public member of its class. The compiler will implicitly *define* A::A() when the compiler uses this *constructor* to create an object of type A .

8. **Express what is meant by java package**

A Package can be defined as a grouping of related types(classes, interfaces, enumerations and annotations) providing access protection and namespace management.

Some of the existing packages in Java are −

java.lang − bundles the fundamental classes

java.io − classes for input , output functions are bundled in this package

9. **Define Static methods**

Static methods are also similar to static variables, you can access them with reference to class name, without creating object. Inside static methods, you cannot access instance variables or instance methods. You can only access static variables or static methods. System.out.println(MyStaticMethods.staticStr);

10. **Express what is the default access to a member in a class**

Default access modifier for class member and member functions is private. Private members are the class members (data members and member functions) that are hidden from the outside world.

11. **Illustrate with example how to import a single package?**

Types of packages in Java: built-in packages and the packages we can create (also known as user defined package).

import java.util.Scanner

Here:

→ java is a top level package

→ util is a sub package

→ and Scanner is a class which is present in the sub package util.

12. Differentiate procedural Vs Object - oriented Programming

Divided Into	In POP, program is divided into small parts called functions.	In OOP, program is divided into parts called objects.
Importance	In POP,Importance is not given to data but to functions as well as sequence of actions to be done.	In OOP, Importance is given to the data rather than procedures or functions because it works as a real world.
Approach	POP follows Top Down approach.	OOP follows Bottom Up approach.
Access Specifiers	POP does not have any access specifier.	OOP has access specifiers named Public, Private, Protected, etc.

13. Explain the features of Java

- * Simple: The Java language is easy to learn. ...
- * Familiar: Java is similar to C/C++ but it removes the drawbacks and complexities of C/C++ like pointers and multiple inheritances. ...
- * Object-Oriented: ...
- * Robust: ...
- * Secure: ...
- * High Performance: ...
- * Multithreaded: ...
- * Platform Independence:

14. Can an inner class declared inside of a method access local variables of this method of this method?

These classes, however, can access the variables or parameters of the block that encloses it only if they are declared as final or are effectively final. ... A local inner class defined inside a method body, have access to it's parameters.

15. What is API Package

Java application programming interface (API) is a list of all classes that are part of the Java development kit (JDK). It includes all Java packages, classes, and interfaces, along with their methods, fields, and constructors.

16. Uses of packages

A package as the name suggests is a pack(group) of classes, interfaces and other packages. In java we use

packages to organize our classes and interfaces. We have two types of packages in Java: built-in packages and the

packages we can create (also known as user defined package).

17. Demonstrate private access specifier

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

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

18. Define constructor

A constructor in Java is a block of code similar to a method that's called when an instance of an object is created. Here are the key differences between a constructor and a method: A constructor doesn't have a return type. The name of the constructor must be the same as the name of the class

19.Define A Java virtual machine (JVM)

It is an implementation of the Java Virtual Machine Specification, interprets compiled Java binary code (called bytecode) for a computer's processor (or "hardware platform") so that it can perform a Java program's instructions.

20.Define encapsulation

It is the process of binding of data and method together in a single entity called class.

PART B

1 Explain the types of package with its importance

i. What is method? How method is defined? give example(6)

2 ii. State the purpose of finalize() method in java? With an example explain how finalize() method can be used in java program(7)

i. What is class? how do you define a class in java(6)

3 ii. Explain the object constructors and calling other constructor with example(6)

4 With relevant examples describe abstraction and encapsulation. Write a java program that uses an abstraction and encapsulation.

5 Illustrate with an example the following features of constructors:

i. Default constructors (2)
ii. Parameterized constructor (2)
iii. Overloaded constructors (2)
iv. A call to another constructor with this operator(2)
v. An object initialization block(2)
vi. A static initialization block(3)

6. i. Illustrate OOPS and explain the features of OOPS (7)

ii. Demonstriate the static felds and methods used in java (6)

UNIT II

PART A

1. Define inheritance hierarchy. Give an example

In object-oriented programming, inheritance enables new objects to take on the properties of existing objects. A class that is used as the basis for inheritance is called a superclass or base class. A class that inherits from a superclass is called a subclass or derived class.

2. How will you define an interface in java program

An interface in the Java programming language is an abstract type that is used to specify a behavior that classes must implement. ... Interfaces are declared using the interface keyword, and may only contain method signature and constant declarations (variable declarations that are declared to be both static and final).

3. What is meant by abstract classes

In Java, we can have an abstract class without any abstract method. This allows us to create classes that cannot be instantiated, but can only be inherited. 4) Abstract classes can also have final methods (methods that cannot be overridden)

4. What is object cloning

Clone() is a method in the Java programming language for object duplication. In java, objects are manipulated through reference variables, and there is no operator for copying an object—the assignment operator duplicates the reference, not the object. The clone() method provides this missing functionality.

5. Define static inner classes

Nested Classes. The Java programming language allows you to define a class within another s. ... A nested class is a member of its enclosing class. Non-static nested classes (inner classes) have ss to other members of the enclosing class, even if they are declared private.

6. In java describe the use of Interfaces?

Interfaces are more flexible, because a class can implement multiple interfaces. Since Java does not have multiple inheritance, using abstract classes prevents your users from using any other class hierarchy. In general, prefer interfaces when there are no default implementations or state.

7. Describe the purpose of the keyword “final”

Final keyword in java. ... First of all, final is a non-access modifier applicable only to a variable, a method or a class. Following are different contexts where final is used. Final variables. When a variable is declared with final keyword, its value can't be modified, essentially, a constant.

8. Describe wrapper classes? Why the wrapper classes are defined as final

Each of Java's eight primitive data types has a class dedicated to it. These are known as wrapper classes because they "wrap" the primitive data type into an object of that class. The wrapper classes are part of the java.lang package, which is imported by default into all Java programs

9.Show how to prevent inheritance

1. Use final.
2. Use private constructors.
3. Use a comment: // do not inherit.
4. Use a javadoc comment.
5. Make every method final, so people can't override them.
6. Use a runtime check in the class constructor: if (this. getClass() != MyClass. class) { throw new RuntimeException("Subclasses not allowed"); }

10. Demonstrate the conditions to be satisfied while declaring abstract

a) **A** class can be marked as abstract without containing any abstract method.
b) A abstract can have one or more abstract method.
c) An abstract method should not have any method body.

11. Illustrate the keyword?usage of super

Java Programming/Keywords/super. It is used inside a sub-class method definition to call a method defined in the super class. ... Only public and protected methods can be called by the super keyword. It is also used by class constructors to invoke constructors of its parent class.

12. Differentiate shallow and deep copy in object cloning

Deep Copy In Java : Deep copy of an object will have exact copy of all the fields of original object just like shallow copy. But in additional, if original object has any references to other objects as fields, then copy of those objects are also created by calling clone() method on them.

13.Distinguish between copying and cloning

This is shallow copy of the object. clone() method of the object class support shallow copy of the object. ... That's why the name shallow copy or shallow cloning in Java. If only primitive type fields or Immutable objects are there then there is no difference between shallow and deep copy in Java.

14.Assess how to reverse ArrayList in Java

You can reverse ArrayList in Java by using the reverse() method of java.util.Collections class. ... By the way, this is a typesafe generic method and you can use it to reverse Integer, String, Float or any kind of List in Java.

15.Deduce the meaning for the keywords : final, finally, finalize

final(lowercase) is a reserved keyword in java. ... The final keyword in java has different meaning depending upon it is applied to variable, class or method. final with Variables : The value of variable cannot be changed once initialized.

16. Create a java program to remove all white spaces from a string in java

a = a.replaceAll("\\s",""); In the context of a regex, \s will remove anything that is a space character (including space, tab characters etc). You need to escape the backslash in Java so the regex turns into \\s . Also, since Strings are immutable it is important that you assign the return value of the regex to a .

17.**Types of exception**

- Arithmetic Exception. It is thrown when an exceptional condition has occurred in an arithmetic operation.
- ArrayIndexOutOfBoundException. ...
- ClassNotFoundException. ...
- FileNotFoundException. ...
- IOException. ...
- InterruptedException. ...
- NoSuchFieldException. ...
- NoSuchMethodException.

18. What is meant by inheritance?

Inheritance is the process by which objects of one class acquire the properties of another class. It supports the concept of hierarchical classification. It provides the idea of reusability. We can add additional features to an existing class without modifying it by deriving a new class from it.

19. What is meant by visibility mode? Mention the visibility modes available.

Visibility mode specifies whether the features of the base class are privately derived or publicly derived. There are three visibility modes. They are,

a. Private
b. Public
c. Protected

20.What are the types in inheritance?

The types in inheritance are,

a. Single inheritance
b. Multiple inheritance c.
Multilevel inheritance d.
Hierarchical inheritance
e. Hybrid inheritance

PART B

1. Define Inheritance? With diagrammatic illustration and Java programs illustrate the different types of inheritance.
2. What is interface? Write a java program to illustrate the use of
3.Write briefly on Abstract classes with an example
4.Describe the sophisticated layout management in user interface component with example
i. Explain the function of object wrapper and auto boxing with suitable example (8)
ii. State the design hints for inheritance(5)
5.Summarize the concept of supper classes and sub classes
6.Illustrate briefly about final methods and classes
i. Demonstrate any six methods available in the StringBuffer class(7)
ii. What is meant by object cloning? Explain it with an example(6)
i. How to define an interface? Why do the members of interface are static and final?(7)
ii. Explain about inner classes and its types with examples(6)
7.Explain the concept of object cloning and inner classes with examples

UNIT III

PART A

1. What are the various traditional error handling methods?

The various traditional error handling methods are,

i. Returning error number.
ii. Global flag manipulation.
iii. Abnormal termination.

2. What is the importance of exceptional handling?

The importance of exceptional handling is,

i. Divide the error handling.
ii. To provide unconditional termination and programmer preferred termination
iii. For separating error reporting and error handling.
iv. To solve the object destroy problem.

3. What are the three keywords used for exception handling mechanism?

The three keywords used for exception handling mechanism are,

i. try → for indicating program area where the exception can be thrown.
ii. throw → for throwing an exception.
iii. catch → for taking an action for specific exception.

4. What is the use of unexpected function?

The unexpected function is called when a function throws an exception not listed in its exception specification. Internally, it calls the terminate function to terminate the program execution. The function set unexpected () is used to call our own unexpected function in place of the built-in unexpected function.

5. What are the challenges in the exception handling mechanism?

The challenges in the exception handling mechanism are,

1. Finding proper catch block.
2. Finding proper catch block for polymorphic object.
3. Backtracking till the beginning of the try block.

6.Why Templates are used in C++?

The Templates are used to develop reusable software component such as functions, classes, etc. Templates allow the construction of a family of templates functions and classes to perform the same operations on different data types.

7. What are rules for invalid function template declaration?

The rules for invalid function template declaration are,

i. No-argument template function.
ii. Template–type argument unused.
iii. Usage of partial number of template arguments.

8. Write the syntax for function Template.

The syntax for function Template is,

```
Template <class T, ……..>
Return Type Fun_Name (arguments)
{
        ………. // body of the template
}
```

9. What are the error handling function in C++?

The error handling function in C++ is,

i. eof()
ii. fail()
iii. bad()
iv. good()

10. What are the rules for virtual function?

The rules for virtual function are,

i. They cannot be static members.
ii. They are access by using object pointers.
iii. A virtual function can be a friend of another class.

11. What are Streams?

Stream is a mechanism, which supplies the input data to a program and presents the processed data in the desired form.

12. What are the file stream classes in C++?

The file stream classes in C++ are,

1. filebuf
2. fstreambase

13.List the different ways to handle exceptions

There are two ways to handle exceptions,

1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions. and

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

14.**Examine the purpose of the finally clause of a try-catch-finally statement**

Finally is usually used to close a code properly after its encountered an Exception or Error. This means that no matter whether an exception/error comes or not, 1nce the try/try-catch block ends, the finally block WILL be executed.

15.Tell the use of assert keyword

Assert is a Java keyword used to define an assert statement. An assert statement is used to declare an expected boolean condition in a program. If the program is running with assertions enabled, then the condition is checked at runtime. If the condition is false, the Java runtime system throws an AssertionError .

16.Compare Input stream and Reader classes

You can wrap an InputStream and turn it into a Reader by using the InputStreamReader class. InputStreams are used to read bytes from a stream. So they are useful for binary data such as images, video and serialized objects. Readers on the other hand are character streams so they are best used to read character data.

17.Compare Input stream and Reader classes

You can wrap an **InputStream** and turn it into a **Reader** by using the **InputStreamReader class**.
tStreams are used to read bytes from a **stream**. So they are useful for binary data such as images, video
serialized **objects**. **Readers** on the other hand are character **streams** so they are best used to read
acter data.

18.Is it necessary that each try block must be followed by a catch block

It is not **necessary that each try block must be followed by a catch block**. It**should be followed** by either a **catch block** or a finally **block**. And whatever exceptions are likely to be thrown **should** be declared in the throws clause of the method.

19. How are the stream classes classified

The order or sequence of bytes in a Java **stream** allow the virtual machine to**classify** it among other **streams**. ... All Java **streams** are derived from Input **Stream**(java.io.InputStream) and Output **Stream**
(java.io.OutputStream) **classes**. They are abstract base **classes** meant for other **stream classes**.

20. Describe runtime exceptions

Definition: An **exception** is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions. When an error occurs within a method, the method creates an object and hands it off to the**runtime** system.

PART – B

1. Define exception. Why it is needed? Explain the different types of exceptions and the exception hierarchy with appropriate examples
2. Explain briefly about user defined exceptions and the concept of throwing and catching exception in java with examples
3. What are input and output streams? Explain them with illustrations Describe the stack trace elements with an example
4. Summarize the concept of streams and stream classes and their Classification
5. the most commonly used classes for handling i/o related Exceptions
6. How exceptions are handled in Java? Explain the important methods
7. While reading a file how would you check whether you have reached
8. Explain how to handle arithmetic exception by giving a suitable

UNIT IV
PART A

1. Identify the different states in thread

A thread can be in any one of the following five states during its lifecycles:

New: A thread is created but didn’t begin its executions.
Runnable: A thread that either is executing at present or that will execute when it gets the access to CPUs.
Terminated: A thread that has completed its executions.
Waiting: A thread that is suspended because it is waiting for some action to occur. For example, it is waiting because of a call to the non-timeout version of *wait()* method or *join()* method or *sleep()*methods.
Blocked: A thread that has suspended executions because it is waiting to acquire a lock or waiting for some I/O event to occur.

2. What do you mean by threads in Java

A **thread**, in the context of **Java**, is the path followed when executing a program.

All**Java** programs have at least one **thread**, known as the main **thread**, which is created by

the **Java** Virtual Machine (JVM) at the program's start, when the main() method is invoked with the

main **thread**.

3.What is multithreading in java

Multithreading in java is a process of executing **multiple threads** simultaneously. Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and**multithreading**, both are used to achieve multitasking.

4.List out the motivation needed in generic programming

In a nutshell, generics enable *types* (classes and interfaces) to be parameters when defining classes, interfaces and methods. Much like the more familiar *formal parameters* used in method declarations, type parameters provide a way for you to re-use the same code with different inputs. The difference is that the inputs to formal parameters are values, while the inputs to type parameters are types.

5.What is meant by notify methods in multithreading

notify() and notifyAll() **methods** with wait() **method** are used to for communication between the threads. ... But when we use notifyAll() **method** then **multiple threads**got the **notification** but execution of threads will be performed one by one because thread requires lock and only one lock is available for one object.

6.Define generic in java

Generics are a facility of generic programming that were added to the Java programming language in 2004 within version J2SE 5.0. They were designed to extend Java's type system to allow "a type or method to operate on objects of various types while providing compile-time type safety"[1]. The aspect *compile-time type safety* was not fully achieved, since it was shown in 2016 that it is not guaranteed in all cases.[2]

7.How to start a thread

The first way is to extend the **Thread** class, override the run() method with the code you want to execute, then **create** a new object from your class and call **start**(). The second method is to pass an implementation of the Runnable interface to the constructor of **Thread**, then call **start**().

8.Describe synchronization in respect to multithreading

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources.Without synchonization, it is possible for one thread to modify a shared variable while another thread is in the process of using or updating same shared variable. This usually leads to significant errors.

9.Why separate wait and sleep methods used in java programming.

Why *Wait* notify *method* is declared in Object Class and not in Thread in *Java* ... that *wait* and notify still remains most confusing for most of *Java programmer* .... to *separate wait*() from *Sleep*() *method* which is also *used* to put Thread on *wait* or ...

10.Develop java interface must be implemented by all threads

All tasks must implement the run method, whether they are a subclass of thread or implement the Runnable interface

11.Difference between suspending and difference stopping a thread in java

Suspending a **thread in java** puts **thread** to sleep indefinitely and it can also be resumed, while on the other hand **stop threading** terminates the **thread** and it can not be resumed. ... Once suspended it is also a simple matter to restart the **thread**.

12.**Why separate wait and sleep methods used in java programming?**

sleep():

It is a static method on Thread class. It makes the current thread into the "Not Runnable" state for specified amount of time. During this time, the thread keeps the lock (monitors) it has acquired.

wait():

It is a method on Object class. It makes the current thread into the "Not Runnable" state. Wait is called on a object, not a thread. Before calling wait() method, the object should be synchronized, means the object should be inside synchronized block. The call to wait() releases the acquired lock.

13.Thread is a light weight process. Comment on this statement

Also threads within a process share the same address space because of which cost of communication between threads is low as it is using the same code section, data section and OS resources, so these all features of thread makes it a "lightweight process".

14.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.

15.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.

16. What if there is a break or return statement in try block followed by finally block?

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.

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.

17.How to create custom exceptions?

By extending the Exception class or one of its subclasses.

<u>Example:</u>

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

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.
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 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.

19) 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.

20) 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.'

PART B

1 Exceptions and the exception hierarchy with appropriate examples
2 Explain briefly about user defined exceptions and the concept of
3 What are input and output streams? Explain them with illustrations
4 Describe the stack trace elements with an example
5 Summarize the concept of streams and stream classes and their Express the most commonly used classes for handling i/o related
6 How exceptions are handled in Java? Explain the important methods used to handle exception
7 While reading a file how would you check whether you have reached the end of the file
8 Explain how to handle arithmetic exception by giving a suitable Example
9 Differentiate byte stream and character stream with necessary Examples
10 Explain the importance of try - catch block with example
11 Evaluate a try block that is likely to generate three types of exception and then incorporate necessary catch blocks and handle them Appropriately
12 Create a new directory by using File object?

UNIT V
PART A
EVENT DRIVEN PROGRAMMING

1) 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.

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 applet?

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 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.

20) 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.

PART B

1. What is layout management? What is the function of Layout manager? (7)

2. What is the process of setting the layout manager(6)
3. Write a program to include the Internal Frame in Swing
4. List the methods available to draw shapes and COLOR
5. Describe in detail about the different layout in Java GUI. Which
6. layout is the default one?
   Summarize the following in detail: Model, view and controller design
7. pattern with respect to Swing in Java. How MVC design pattern is achieved?
8. Discuss mouse listener and mouse motion listener. Give an example Program
9. Demonstrate the Characteristics of model view Controller design patterns and its advantages
   Illustrate the usage of special fonts for text in graphics programming
   i. Clasify the classes under 2D shapes (7)
   ii. Explain the Swing components in detail(6)
   iii. Infer JList and JComboBox with an example(7)
   iv. Compare check boxes and radio buttons with an example(6)