CS8392 - OOPS

UNIT I – INTRODUCTION TO OOP AND FUNDAMENTALS OF JAVA

Part A – Question Bank

1. **Define OOP.**

   Object-Oriented Programming (OOP) is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

   - Object
   - Class
   - Inheritance
   - Polymorphism
   - Abstraction
   - Encapsulation

2. **Define object and class.**

   Any entity that has state and behavior is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be either physical or logical.

   A class is the basic building block of an object-oriented language. It is a template that describes the data and behavior associated with instances of that class. The data associ- ated with a class or object is stored in variables and the behavior associated with a class or object is implemented with methods.

3. **How can we create an instance of a class in java?**

   To create an instance of a class:

   - Declare an instance identifier (instance name) of a particular class.
   - Construct the instance (i.e., allocate storage for the instance and initialize the instance) using the “new” operator.

4. **Define Inheritance.**

   When one object acquires all the properties and behaviours of parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

5. **What are the types of inheritance in java?**

   1. Single Inheritance
   2. Multilevel Inheritance
   3. Hierarchical Inheritance
   4. Hybrid Inheritance

6. **Define Polymorphism.**

   Polymorphism means taking many forms, where „poly‟ means many and „morph‟ means forms. It is the ability of a variable, function or object to take on multiple forms.

7. **Define abstraction.**

   Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don‟t know the internal processing.

8. **Define encapsulation.**

   - Binding (or wrapping) code and data together into a single unit is known as encapsulation. For example: capsule, it is wrapped with different medicines.
   - A java class is the example of encapsulation.

9. **List the advantage of OOPs over Procedure-oriented programming language**

   OOPs makes development and maintenance easier. But in Procedure-oriented program- ming language, it is not easy to manage if code grows as project size grows.

   OOPs provides data hiding whereas in Procedure-oriented programming language, global data can be accessed from anywhere.

   OOPs provides ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.

10. **What is difference between object-oriented programming language and object-based programming language?**

Object oriented language

- Object-oriented language supports all the features of OOPs.
- Object-oriented language doesn‟t has in-built object.
- Object-oriented languages are C++, C#, Java etc.

Object based language

- Object-based language doesn‟t support all the features of OOPs like Polymorphism and Inheritance
- Object-based language has in-built object like javascript has window object.
- Object-based languages are Javascript, VB etc.

11. **What are the three major sections of java source file?**

    The source consists of three major sections:

    - The package
    - The import
    - Class definition

12. **List out the source file declaration rules.**

    - There can be only one public class per source file.
    - A source file can have multiple non-public classes.
    - The public class name should be the name of the source file which should have .java extension at the end.

- For eg, if the class name is public class Employee{}, then the source file should be as Employee.java.

- If the class is defined inside a package, then the package statement should be the first statement in the source file.
- If import statements are present, then they must be written between the package statement and the class declaration. If there are no package statements, then the import statement should be the first line in the source file.
- Import and package statements will imply to all the classes present in the source file. It is not possible to declare different import and/or package statements to different classes in the source file.

13. **Define Jvm.**

The JVM is an interpreter for the bytecode form of the program. It steps through one byte- code instruction at a time. It is an abstract computing machine that enables a computer to run a Java program.

14. **What is bytecode?**

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

15. **Write a note on integer data types in java.**

Integers are used for storing integer values. There are four kinds of integer types in Java. Each of these can hold a different range of values. The values can either be positive or negative.

type	size
byte	8 bits
short	16 bits
int	32 bits
long	64 bits

16. **Write a note on float data types in Java.**

Float is used to store numbers with decimal part. There are two floating point data types in Java namely, the float and the double.

type	size
float	32 bits
double	64 bits

17. **give any three OOP concepts.**

- Encapsulation
- Inheritance
- Polymorphism

18. **Write a note on import statement?**

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

```
import <classname>
```

The whole path of the class must be specified to import a class from the Java library, For instance, to import the Date class from the util package, the following code is used:

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

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

19. List out the features of java.

- Simple
- Secure
- Portable
- Object-oriented
- Robust
- Multithreaded
- Architecture-neutral
- Interpreted
- High performance
- Distributed
- Dynamic

20. What is the use of comment?

The contents of a comment are ignored by the compiler. Instead, a comment can be used to describe or explain the operation of the program to anyone who is reading its source code.

21. What is a variable? How to declare variable in java?

The variable is the basic unit of storage in a java program. A variable is defined by the combination of an identifier, a type, and an optional initialize. All variables must be declared before they can be used. The basic form of a variable declaration is shown have

```
Type identifier [= value],[,identifier [=value]]
```

The type in one ofjava‟satomic types.The identifieristhe nameofthe variable.For example

```
int a,b,c;
int d=3,c=5;
```

22. What is a variable? What are the different types of variables?

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

- Instance variable
- Local variable
- Class variable

23. **What are the difference between static variable and instance variable?**

The data or variables, defined within a class are called instance variables.Instance variables declared as static are, essentially, global variables. When objects of its class are declared, no copy of a static variable is made.

24. **Write a note on conditional operator in java.**

The conditional operator is otherwise known as the ternary operator and is considered to be an alternative to the if else construct. It returns a value and the syntax is:

<test> ? <pass> : <fail>

Where,<test> is the condition to be tested. If the condition returns true then the state- ment given in <pass> will be executed. Otherwise, the statement given in <fail> will be executed.

25. **List out the operator in java**

- Arithmetic Operators
- Increment and Decrement Operators
- Bitewise Operators
- Relational Operators
- Logical Operators
- Assignment Operators

26. **What are jump statements in java?**

In java have three jump statements

- return
- continue
- break

27. **differentiable between break and continue statements?**

The break keyword halts the execution of the current loop and forces control out of the loop. The term break refers to the act of breaking out of a b lock of code. Continue is similar to break, except that instead of halting the execution of the loop, it starts the next iteration.

28. **What is a class? give an example?**

A class defines the shape and behavior of an object and is a template for multiple objects with similar features.

or

A class is a new data type. Once defined, this new type can be used to create ob- jects of that type. Thus, a class is a template for an object, and an object is an instance of a class.

29. **What are constructors?**

A constructor initializes an object immediately upon creation. It has the same name as the class in which it resides and is syntactically similar to a method. Once defined, the con- structor is automatically called immediately after the object is created, before the ***new*** op- erator completes.

30. **What's the difference between constructors and other methods?**

Constructors must have the same name as the class and cannot return a value. They are only called once while regular methods could be called many times.

31. **What is a package?**

A Package is a container that groups related types (classes, interfaces, enumerations and annotations). It is similar to folders in computer. It is generally used to control access and to avoid naming collision. The syntax for creating package is

Syntax:

```
package pkg_name;
```

32. **What is static variable?**

Variable declared with keyword *static* is a static variable. It is a class level variable com- monly shared by all objects of the class.

- Memory allocation for such variables only happens once when the class is loaded in the memory.
- scope of the static variable is class scope (accessible only inside the class)
- lifetime is global (memory is assigned till the class is removed by JVM).
- Automatically initialized to 0.

Example:

```
static int no;
```

33. **Write short notes on static method.**

The method declared with static keyword is known as static method. *main()* is most com- mon static method.

- It belongs to the class and not to object of a class.
- A static method can directly access only static variables of class and directly invoke only static methods of the class.
- It can be called through the name of class without creating any instance of that class. For example, ClassName.methodName()

Example:

```
static void show(){
        System,out.println("Hello");
```

}

34. **What do you mean by static import?**

The static import allows the programmer to access any static members of imported class directly. There is no need to qualify it by its name. *Syntax:*

```
import static package_name;
```

35. **What is a static block?**

A static block is a block of code enclosed in braces, preceded by the keyword *static*.

- The statements within the static block are first executed automatically before main when the class is loaded into JVM.
- A class can have any number of static blocks.

PART-B

1. Explain about packages. Give an example program which uses packages. (16) (CS1261 NOV /DEC 2016) (IT2301 NOV/DEC-2012)
2. Explain with the help of a program how object oriented programming overcomes the shortcomings of procedure oriented programming.(8) (IT2301 APR/MAY-2015)
3. Given two one dimensional arrays A and B which are sorted in ascending order. Write a Java program to merge them into a single sorted array, see that is contains every item from array A and B, in ascending order. (8) (IT2301 APR/MAY-2015)
4. With an example, describe in detail about how polymorphism plays a useful role in Java. (8) (IT2301 APR/MAY-2015)
5. Elaborate on the various object oriented concepts, with necessary illustrations. (16) (IT2301 MAY/JUNE-2014)
6. Write a program to perform the following functions using classes, objects, constructors and destructors where essential. (IT2301 MAY/JUNE-2014)
   a. Get as input the marks of 5 students in 5 subjects. (4)
   b. Calculate the total and average. (8)
   c. Print the formatted result on the screen. (4)
7. With suitable examples explain how packages can be created, imported and used. Also elaborate on its scope. (16) (IT2301 MAY/JUNE-2014)
8. Write a program to perform following functions on a given matrix.

   (IT2301 MAY/JUNE-2014)
   a. Find the row and column sum. (8)
   b. Interchange rows and columns. (8)
9. Describe the structure of Java program. (8) (IT2301 NOV/DEC-2012)
10. Explain the features of Java. (8) (IT2301 NOV/DEC-2012)
11. Write a simple Java program to implement basic Calculator operations.(16)

    (CS2311 NOV/DEC-2014)

12. How packages are used to resolve naming conflicts in Java? With an example show to add classes to packages and how to import packages in classes. (16)

    (CS2311 NOV/DEC-2014)

13. Write a program in Java that interchanges the odd and even components of an array in
    Java. (16) (CS2311 APR/MAY-2015)

14. Write a Java program to sort set of names stored in an array in alphabetical order. (16) (CS2311 APR/MAY-2015)

15. Explain the arrays and its types in detail with example program.

    (CS2311 MAY/JUNE-2014)

16. Briefly explain about the key elements of Object Oriented Programming.(16)

    (CS1361 NOV/DEC-2014)

17. Explain about Class, Objects and Methods in Java with an example program. (16)

    (CS1361 NOV/DEC-2014)

18. Describe the following: (CS1361 NOV/DEC-2014)
    a. Features of Java (8)
    b. Data types in Java (8)

19. Explain about Package in Java. List built in Java API packages. (16)

    (CS1361 NOV/DEC-2014)

20. Discuss the various parameter passing methods in Java. Illustrate with examples.

    (CS1361 NOV/DEC-2014)

21. Explain the arrays and its types in detail with example program.

    (CS2311 MAY/JUNE-2014)

22. Briefly explain about the key elements of Object Oriented Programming.(16)

    (CS1361 NOV/DEC-2014)

23. Explain about Class, Objects and Methods in Java with an example program. (16)

    (CS1361 NOV/DEC-2014)

24. Describe the following: (CS1361 NOV/DEC-2014)
    a. Features of Java (8)
    b. Data types in Java (8)

25. Explain about Package in Java. List built in Java API packages. (16)

    (CS1361 NOV/DEC-2014)

26. Discuss the various parameter passing methods in Java. Illustrate with examples.

    (CS1361 NOV/DEC-2014)

UNIT II – INHERITANCE AND INTERFACE

Part A – Question Bank

1. What is finalize () method?

The finalize() method is called just before an object is garbage collected. It is used to dispose system resources, perform clean-up activities and minimize memory leaks.

Syntax:

```
protected void finalize()                    // finalize() is called just once on an object
{
    ………………
}
```

2. What is meant by Inheritance?

Inheritance is the mechanism in java by which one class is allow to inherit the features (fields and methods) of another class. It is process of deriving a new class from an exist- ing class.

Syntax:

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

3. list out the types of inheritance.

- □ Single inheritance
- □ Multilevel inheritance
- □ Multiple inheritance
- □ Hybrid inheritance
- □ Hierarchical inheritance

4. how do you implement multiple inheritance in java?

Java does not allow multiple inheritance:

- □ To reduce the complexity and simplify the language
- □ To avoid the ambiguity caused by multiple inheritance

It can be implemented using Interfaces.

5. Define super class and subclass.

The process of deriving a new class from an existing class is inheritance. A class that is inherited is called a *superclass* and the class that does the inheriting is called a *subclass.*

6. state the use of keyword super.

- □ It an be used to refer immediate parent class instance variable when both parent and child class have member with same name
- □ It can be used to invoke immediate parent class method when child class has overridden that method.
- □ super() can be used to invoke immediate parent class constructor.

7. What is the use of 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.

8. Write short notes on Object class.

The Object class is the parent class of all the classes in java by default (directly or indirectly). The java.lang.Object class is the root of the class hierarchy. Some of the Object class are Boolean, Math, Number, String etc

9. Define abstract class?

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

10. When to use abstract Methods & abstract class?

Abstract methods are usually declared where two or more subclasses are expected to do a similar thing in different ways through different implementations. These subclasses extend the same Abstract class and provide different implementations for the abstract methods.

Abstract classes are used to define generic types of behaviors at the top of an object- oriented programming class hierarchy, and use its subclasses to provide implementation details of the abstract class.

11. List out the rules in defining abstract classes.

- □ An abstract class may have concrete (complete) methods.
- □ An abstract class may or may not have an abstract method. But if any class has one or more abstract methods, it must be compulsorily labeled abstract.
- □ Abstract classes can have Constructors, Member variables and Normal methods.
- □ Abstract classes are never instantiated.
- □ For design purpose, a class can be declared abstract even if it does not contain any abstract methods.
- □ Reference of an abstract class can point to objects of its sub-classes thereby achieving run-time polymorphism Ex: Shape obj = new Rectangle();
- □ A class derived from the abstract class must implement all those methods that are declared as abstract in the parent class.
- □ If a child does not implement all the abstract methods of abstract parent class, then

the child class must need to be declared abstract as well.

12. Define final class.

When a class is declared with *final* keyword, it is called a final class. A final class cannot be extended (inherited).

13. What are the uses of a final class?

- To prevent inheritance, as final classes cannot be extended. For example, all Wrapper Classes like Integer, Float etc. are final classes. We cannot extend them.
- To create an immutable class like the predefined String class. We cannot make a class immutable without making it final.

14. What are the benefits of final keyword in Java?

- Final keyword improves performance. Not just JVM can cache final variable but also application can cache frequently use final variables.
- Final variables are safe to share in multi-threading environment without additional synchronization overhead.
- Final keyword allows JVM to an optimized method, variable or class.

15. Is final method inherited?

Yes, final method is inherited but we cannot override it.

For Example:

```
class Bike{ final void
  run()
{
System.out.println("running...");
}
}
public class Honda2 extends Bike
{
   public static void main(String args[]){
    new Honda2().run();
}
}
```

16. What is blank or uninitialized final variable?

A final variable that is not initialized at the time of declaration is known as blank final variable.

If we want to create a variable that is initialized at the time of creating object and once initialized may not be changed, it is useful.

It can be initialized only in constructor.

17. Define static blank final variable.

A static final variable that is not initialized at the time of declaration is known as static blank final variable. It can be initialized only in static block.

Example of static blank final variable

```
      public class A{
  static final int data;//static blank final variable  static{
         data=50;}
         public static void main(String args[]){
           System.out.println(A.data);
 }
      }
```

18. What is meant by interface?

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types.

19. 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, we have to inherit our class from it and Java does not allow multiple inheritance. On the other hand, we can implement multiple interfaces in your class.

20. What are the uses of interface?

- □ Interfaces are used to implement abstraction.
- □ Since java does not support multiple inheritance in case of class, it can be achieved by using interface.
- □ It is also used to achieve loose coupling.

21. Define nested interface.

An interface which is declared inside another interface or class is called nested inter- face. They are also known as inner interface. Since nested interface cannot be accessed directly, the main purpose of using them is to resolve the namespace by grouping related interfaces (or related interface and class) together.

22. How will you find out the length of a string in java? Give an example?

The length ( ) method is used to number of characters is string. For example,

String str="Hello";

System.out.println("Length of string is "+str.length( ));

23. you are planning to do an indexed search in a list of objects. Which of the two java collections should you use: arraylist or linkedlist?

ArrayList

24. how would you make a copy of an entire java object with its state?

Have this class implement Cloneable interface and call its method clone().

25. What is the basic difference between string and stringbuffer object? –

String is an immutable object. StringBuffer is a mutable object.

26. What is inner class?

If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.

27. 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 inter- face. 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.

28. What would you use to compare two string variables - the operator == or the method equals()?

I'd use the method equals() to compare the values of the Strings and the = = to check if two variables point at the same instance of a String object.

29. can an inner class declared inside of method access local variables of this method?

It's possible if these variables are final.

30. What's the main difference between a vector and an arraylist ?

Java Vector class is internally synchronized and ArrayList is not.

PART-B

1. **Explain overriding methods and final methods in Java.(8) (CS1261 MAY /JUNE 2016)**
2. Create a Java class Shape with constructor to initialize the one parameter "dimension". Now create three sub classes of Shape with following methods: (16)

   (IT2301 APR/MAY-2015)

   a. "Circle" with methods to calculate the area and circumference of the circle with di- mension as radius
   b. "Square" with methods to calculate the area and length of diagonal of the square with dimension as length of one side.
   c. "Sphere" with methods to calculate the volume and surface area of the sphere with dimension as radius of the sphere. Write appropriate main method to create object of each class and test every method.
3. Explain with example how multiple inheritance is achieved in Java. (16)

   (IT2301 APR/MAY-2015)
4. What are interfaces? Explain with an example how multiple inheritance is

implemented using interfaces. (16) (CS2311 NOV/DEC-2014)

5. Develop a Library interface which has drawbook(), returnbook() (with fine), checksta- tus() and reservebook() methods. All the methods are tagged with public in the following ways: (16) (IT2301 APR/MAY-2015)
   a. Using draw book() - get the required book based on title
   b. Using checkstatus – user book returned date details
   c. Using with fine() – whether failed to return the book within a time period charge –
      Rs.5/day
   d. Using reserve book() – block or reserve particular book for their account.
6. Explain about inheritance in Java. (16) (IT2301 NOV/DEC-2012) (CS1361 NOV/DEC- 2014)
7. Explain the interfaces in detail with suitable example. (CS2311 MAY/JUNE-2014)

UNIT III – EXCEPTION HANDLING AND I/O

Part A – Question Bank

1. What is 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 OutOf- MemoryError, which would be reported by the JVM itself.

3. Which is super class 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 manage- ment 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 excep- tions.

- Checked exceptions: Achecked 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?

An 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 excep- tion. Checked exceptions must be caught using try catch () block or we should throw the exception using throws clause. If you don't, 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. Excep- tions 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 finallyclause.

13. 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.

14. Write short notes on throwable class.

The Throwable class is the superclass of all errors and exceptions in the Java language. It contains a snapshot of the execution stack of its thread at the time it was created. It can also contain a message string that gives more information about the error. The Exception class inherits all the methods from the class Throwable.

15. Write short notes on Exception class.

Java allows the user to create their own exception class which is derived from built-in class Exception. The Exception class inherits all the methods from the class Throwable.

Two commonly used constructors of Exception class are:

- Exception() - Constructs a new exception with null as its detail message.
- Exception(String message) - Constructs a new exception with the specified detail message.

Syntax:

```
class User_defined_name extends Exception{
……….
}
```

16. How is custom exception created?

The user defined exception is created by extending the Exception class or one of its sub- classes.

Example:

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

17. What are the different ways to handle exceptions?

There are two ways to handle Exceptions

- Wrapping the desire code in a try block followed by a catch block to catch the exception
- List the desired exception in the throws clause of the method and let the caller of the method handle those exceptions.

18. What do you mean by chained exceptions?

Chained Exceptions allows to relate one exception with another exception, i.e one exception describes cause of another exception.

Throwable methods that supports chained exceptions are:

1. getCause() method :- This method returns actual cause of an exception.
2. initCause(Throwable cause) method :- This method sets the cause for the calling exception.

19. Write short notes on stacktraceElement.

The StackTraceElement class element represents a single stack frame which is a stack trace when an exception occurs. Extracting stack trace from an exception could provide useful information such as class name, method name, file name, and the source-code line number. This creates a stack trace element representing the specified execution point.

getStackTrace( ) - returns an array of StackTraceElements

20. list any three common run time errors.

Exception	Description
ArithmeticException	Arithmetic error, such as divide-by-zero
ArrayIndexOutOfBoundsException	Array index is out-of-bounds
IllegalThreadStateException	Requested operation not compatible with current thread state

21. Differentiate throw and throws in Java.

throw	throws
Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.
Throw is followed by an instance.	Throws is followed by class.
Throw is used within the method.	Throws is used with the method signa-ture.
You cannot throw multiple exceptions.	You can declare multiple exceptions e.g. public void method()throws IO Exception, SQL Exception.
void m(){ throw new IOException("Error"); }	void m() throws IOException{ //code }

22. Define Stream.

A stream can be defined as a sequence of data. The InputStream is used to read data from a source and the OutputStream is used for writing data to a destination.

23. list the type of streams supported in Java.

1. ***Byte Stream :*** It is used for handling input and output of 8 bit bytes. The frequently used classes are FileInputStream and FileOutputStream.
2. ***Character Stream :*** It is used for handling input and output of characters. Character stream uses 16 bit Unicode. The frequently used classes are FileReader and File Writer.

24. List out the predefined streams available in Java.

Java provides the following three standard streams −

- Standard Input − refers to the standard InputStream which is the keyboard by default. This is used to feed the data to user's program and represented as **system.in**.
- Standard Output − refers to the standard OutputStream by default,this is console and represented as **system.out**.
- Standard Error − This is used to output the error data produced by the user's program and usually a computer screen is used for standard error stream and

represented as **system.err**.

25. What is Byte stream in Java? list some of the Bytestream classes.

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

List of Byte Stream classes

- Byte Array Input Stream
- Byte Array Output Stream
- Filtered Byte Streams
- Buffered Byte Streams

26. What is character stream in Java? list some of the characterstream classes.

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

List of Character Stream classes

- File Reader
- File Writer
- Char Array Reader
- Char Array Writer

27. What are the steps involved in stream i/ooperations?

1. Open an input/output stream associated with a physical device (e.g., file, network, console/keyboard), by constructing an appropriate I/O stream instance.
2. Read from the opened input stream until "end-of-stream" encountered, or *write* to the opened output stream (and optionally flush the buffered output).
3. Close the input/output stream.

28. Write about read() method.

- returns the input byte read as an int in the range of 0 to 255, or
- returns -1 if "end of stream" condition is detected, or
- throws an IOException if it encounters an I/O error.
- The read() method returns an int instead of a byte, because it uses -1 to indicate end- of-stream.
- The read() method blocks until a byte is available, an I/O error occurs, or the "end-of-stream" is detected.

29. What are the two variations of read() method?

public int read(byte[] bytes, int offset, int length) throws IOException

public int read(byte[] bytes) throws IOException

30. What are the two variations of write() method?

public void write(byte[] bytes, int offset, int length) throws IOException public void write(byte[] bytes) throws IOException

31. What's the difference between println(), print() and printf()?

print()	-	prints string inside the quotes
println()	-	prints string inside the quotes similar like print() method. Then the cursor moves to the beginning of the next line.
printf()	-	it provides string formatting (similar to printf in C/C++ programming).

32. Write about Fileinputstream.

- This stream is used for reading data from the files.
- Objects can be created using the keyword new and there are several types of constructors available.
- The two constructors which can be used to create a FileInputStream object:

i) Following constructor takes a file name as a string to create an input stream object to read the file:

```
OutputStream f = new FileOutputStream("filename ");
```

ii) Following constructor takes a file object to create an input stream object to read the file. First we create a file object using File() method as follows:

```
File f = new File("filename ");
InputStream f = new FileInputStream(f);
```

33. Write about Fileoutputstream.

- FileOutputStream is used to create a file and write data into it.
- The stream would create a file, if it doesn't already exist, before opening it for output.
- The two constructors which can be used to create a FileOutputStream object:

i) Following constructor takes a file name as a string to create an input stream object to write the file:

```
OutputStream f = new FileOutputStream("filename ");
```

ii) Following constructor takes a file object to create an output stream object to write the file. First, we create a file object using File() method as follows:

```
File f = new File("filename ");
OutputStream f = new FileOutputStream(f);
```

ARUNAI ENGINEERING COLLEGE, TIRUVANNAMALAI

PART -B

1. Explain exception handling in Java with an example. (8)(CS1261 MAY /JUNE 2016)
2. Explain with an example the exception handling feature in Java.(8)

   (CS1261 NOV /DEC 2016) (CS2311 APR/MAY-2015)
3. Explain I/O streams with suitable examples. (8) (IT2301 APR/MAY-2015)
4. Discuss about the Java error handling mechanism? What is the difference between "un- checked exceptions" and "checked exceptions"? What is the implication of catching all the exceptions with the type "Exception"? (8) (IT2301 APR/MAY-2015)
5. How are exceptions handled in Java? Elaborate with suitable examples. (8)

   (IT2301 MAY/JUNE-2014)
6. Describe about different input and output streams and their classes. (16)

   (IT2301 NOV/DEC-2012)
7. Discuss about throwing and catching exceptions. (16) (IT2301 NOV/DEC-2012)
8. Explain the concept of streams and its byte stream classes in detail.

   (CS2311 MAY/JUNE-2014)
9. Explain the use of File stream classes and file modes. Write an example program for file manipulation. (16) (CS1361 NOV/DEC-2014)
10. Describe different types of exceptions. Write a program for handling Array out of bounds Exception. (16) (CS1361 NOV/DEC-2014)
11. Write a Java program that asks the user for a number and displays ten times the number. Also the program must throw an exception when the user enters a value greater than 10. (16) (CS1361 MAY/JUNE-2013)

UNIT –IV

1. What are the benefits of Multithreading?

- Multithreading increases the **responsiveness** of system as, if one thread of the application is not responding, the other would respond in that sense the user would not have to sit idle.
- Multithreading allows **resource sharing** as threads belonging to the same process can share code and data of the process and it allows a process to have multiple threads at the same time active in **same address space**.
- Creating a different process is costlier as the system has to allocate different memory and resources to each process, but creating threads is easy as it does not require allocating separate memory and resources for threads of the same process.

2. What are the differences between Multithreading and Multitasking?

Parameter	Multithreading	Multitasking
Definition	Multithreading is to execute multiple threads in a process concurrently.	Multitasking is to run multiple processes on a computer concurrently.
execution	In Multithreading, the CPU switches between multiple threads in the same process.	In Multitasking, the CPU switches between multiple processes to complete the execution.
resource sharing	In Multithreading, resources are shared among multiple threads in a process.	In Multitasking, resources are shared among multiple processes.
Complexity	Multithreading is light-weight and easy to create.	Multitasking is heavy-weight and harder to create.

3. Define thread.

A thread is a basic execution unit which has its own program counter, set of the register and stack. But it shares the code, data, and file of the process to which it belongs.

4. list the states of thread life cycle.

A thread at any point of time exists in any one of the following states.

1. New
2. Runnable
3. Blocked
4. Waiting
5. Timed Waiting
6. Terminated

5. list some methods supported by threads.

- ***join():*** It makes to wait for this thread to die. We can wait for a thread to finish by calling its join() method.

- ***sleep():*** It makes current executing thread to sleep for a specified interval of time. Time is in milli seconds.
- *yield():* It makes current executing thread object to pause temporarily and gives control to other thread to execute.
- ***notify():*** This method is inherited from Object class. This method wakes up a single thread that is waiting on this object's monitor to acquire lock.
- ***notifyAll():*** This method is inherited from Object class. This method wakes up all threads that are waiting on this object's monitor to acquire lock.
- ***wait():*** This method is inherited from Object class. This method makes current thread to wait until another thread invokes the notify() or the notifyAll() for this object.

6. Can we start a thread twice?

No. After starting a thread, it can never be started again. If we do so, an Illegal Thread-State Exception is thrown. In such case, thread will run once but for second time, it will throw exception.

7. What is the use of join() method?

The join() method waits for a thread to die. In other words, it causes the currently running threads to stop executing until the thread it joins with completes its task.

Syntax:

```
public void join()throws InterruptedException
public void join (long milliseconds) throws Interrupted Exception
```

8. Why do we need Synchronization?

The synchronization is mainly used to

i) To prevent thread interference.

ii) To prevent consistency problem.

9. What are the types of Synchronization?

There are two types of synchronization:

i) Process Synchronization

ii) Thread Synchronization

10. Write about Java synchronized method.

- If we declare any method as synchronized, it is known as synchronized method.
- Synchronized method is used to lock an object for any shared resource.
- When a thread invokes a synchronized method, it automatically acquires the lock for that object and releases it when the thread completes its task.

11. What do you mean by Synchronized block in java?

Synchronized block can be used to perform synchronization on any specific resource of the method. Suppose we have 50 lines of code in your method, but we want to synchronize only 5 lines, we can use synchronized block.

If we put all the codes of the method in the synchronized block, it will work same as the synchronized method.

12. What is synchronization? Briefly explain.

Two or more threads accessing the same data simultaneously may lead to loss of data integrity. For example, when two people access a savings account, it is possible that one person may overdraw and the cheque may bounce. The importance of updating of the pass book can be well understood in this case.

13. Why would you use a synchronized block vs. synchronized method?

Synchronized blocks place locks for shorter periods than synchronized methods.

14. What's the difference between the methods sleep() and wait()

The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the no- tify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

15. 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 updat- ing that object's value. This often leads to significant errors.

16. 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() meth- od.

17. 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.

18. 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.

19. 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.

20. Can I implement my own start() method?

The Thread start() method is not marked final, but should not be overridden. This meth- od 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.

21. 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.

22. 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.

23. 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.

24. How would you implement a thread pool?

The Thread Pool class is a generic implementation of a thread pool, which takes the fol- lowing 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 ac- tive threads that are waiting for ctivation. once the threads have finished processing they come back and wait once again in the pool.

25. What is Generics? List its advantages.

Generics are similar to templates in C++. Using Generics, the programmer can create a single class, interface or method that automatically works with all types of data (Integer, String, Float etc) i.e. the code can be reused for object of many different types.

Advantages of Generics:

- Generics make the code safer and easier to read.
- Type casting is not needed.
- It is checked at compile time. So problem will not be generated at run time.

26. How to implement generic method in Java?

Generic methods are methods that can accept any type of argument.

Syntax:

```
<type-parameter> return_type method_name (parameters) {...}
```

Example:

```
public void display(T data) {
      System.out.println(data);
}
```

27. What is the question mark in Java Generics used for?

The wildcard element is represented by ? and it specifies an unknown type i.e. it means any type. For example,

<? extends Number>

specifies any child classes of class Number e.g. Integer, Float, double etc.

28. What do you mean by bounded type parameters?

In Generics, bounded type parameter set restriction on the type that will be allowed to pass to a type-parameter.

Syntax: <T extends superclass>

For example, <T extends Number>

In this example, the type parameter is restricted to any child classes of class Number e.g. Integer, Float, double etc.

29. What is Erasure?

The Java compiler applies type erasure to implement generic programming. It replaces all type parameters in generic types with their bounds or Object if the type parameters are unbounded.

Example: Java class with Generics

```
class Gen<T>{
        T get(){
        …}
}
```

Example: T replaced by java.lang.Object

```
class Gen extends java.lang.Object{
        java.lang.Object get(){
        …}
}
```

30. list out the restrictions to be considered in generic programming.

- 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 instanceof With Parameterized Types
- Cannot Create Arrays of Parameterized Types
- Cannot Create, Catch, or Throw Objects of Parameterized Types

Part-B

1. With a simple program explain the multithreading concept in Java.(16)

   (CS1261 NOV /DEC 2016) (CS2311 APR/MAY-2015)

2. Write a complex program to illustrate about thread priorities. Imagine that the first thread has just begun to run, even before it has a chance to do anything. Now the higher priority thread that wants to run as well. Now the higher priority thread has to do its work before the first thread starts. (16). (IT2301 APR/MAY-2015)

3. Explain life cycle of a thread with help of a diagram. How do the wait and notify all/no-tify methods enable cooperation between thread? (8) (IT2301 APR/MAY-2015)

4. Explain in detail about generic method with a suitable example. (8)

   (IT2301 APR/MAY-2015)

5. With illustrations explain multithreading, interrupting threads, thread states and thread priorities. (16) (IT2301 MAY/JUNE-2014)

6. What is a Thread? What are the different states of Thread? Explain the creation of Thread with an example program. (16) (CS1361 NOV/DEC-2014)

7. Using generic classes, write a program to perform the following operations on an array.

   (IT2301 MAY/JUNE-2014)

   a. Add an element in the beginning/middle/end. (8)
   b. Delete an element from a given position. (8)

8. What are interrupting threads? Explain thread states and synchronization. (16)

   (IT2301 NOV/DEC-2012)

9. What is multithreading? Explain the two methods of implementing threads with an ex- ample. (16)(CS2311 NOV/DEC-2014)

10. Explain the lifecycle of a thread in detail with example program.

    (CS2311 MAY/JUNE-2014)

UNIT-V

TWO MARKS QUESTION & ANSWER

1. What is the difference between the 'Font' and 'FontMetrics' class?

Font class is used to set or retrieve the screen fonts.The Font class maps the characters of the language to their respective glyphs.

The FontMetrics class defines a font metrics object, which encapsulates information about the rendering of a particular font on a particular screen.i.e they provide access to the attributes of the font object.

2. What is the difference between the paint() and repaint() methods?

paint()	repaint()
The paint() method is called when some action is performed on the window.	Whenever a repaint method is called, the update method is also called along with paint() method.
This method supports painting via graphics object.	This method is used to cause paint() to be invoked by the AWT painting thread.

3. What is the difference between applications and applets?

Java Application	Applet
Java application on contains a main method	An applet does not contain a main method
Does not require internet connection to execute	Requires internet connection to execute
Is stand alone application	Is a part of web page
Can be run without a browser	Requires a Java compatible browser
Users stream I/O classes	Use GUI interface provided by AWT or Swings
Entry point is main method	Entry point is init method
Generally used for console programs	Generally used for GUI interfaces

4. 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.

5. 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, select- ing a list, etc. There are two types of models for handling events and they are: a) event- inheritance model and b) event-delegation model

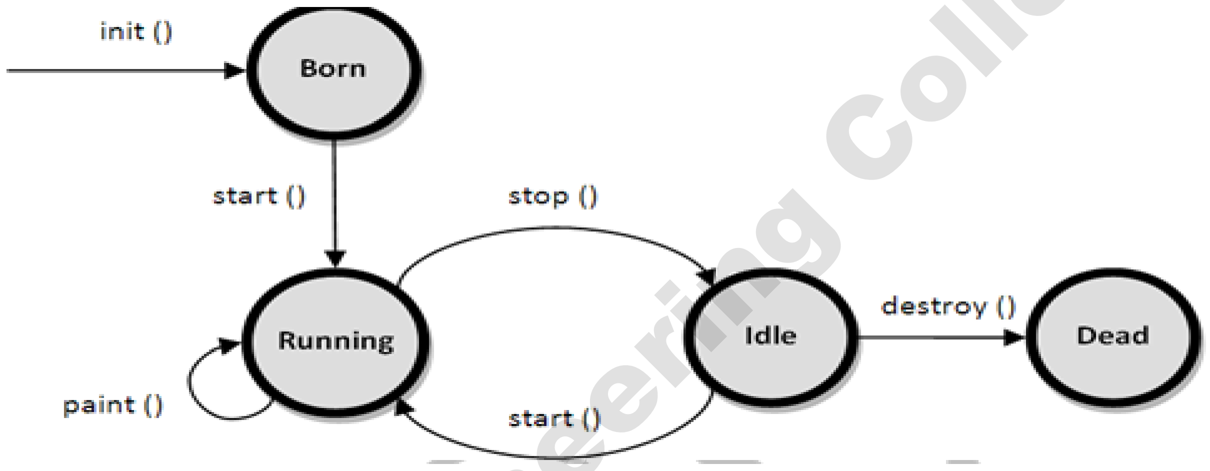
6. What is the lifecycle of an applet?

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.



7. 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 hierar- chy.

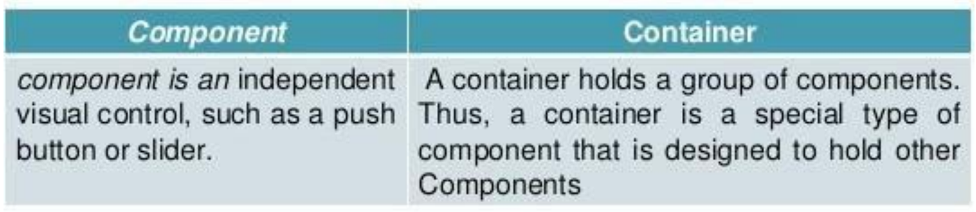
8. Write a note on Borderlayout?

The BorderLayout class implements a common layout style for top-level windows. It has four narrow, fixed-width components at the edges and one large area in the center. The four sides are referred to as north, south, east, and west. The middles area is called the center. Here are the constructors defined by BorderLayout

BorderLayout( ) BorderLayout(int horz, int vert)

9. Distinguish between component and container

Java's Component class represents visual elements of a Graphical User Interface. Its sub- classes include Button, Checkbox, TextField, Choice, and Canvas. The Container class is another subclass of Component. A Container is a component that can contain other components (including other containers). This is the essential difference between con- tainers and other types of component. Subclasses of Container include Frame, Panel, and Applet.



Component	Container
component is an independent visual control, such as a push button or slider.	A container holds a group of components. Thus, a container is a special type of component that is designed to hold other Components

10. What is the difference between jcheckbox and jradiobutton?

In a checkbox group, a user can select more than one option. Each checkbox operates individually, so a user can toggle each response “on” and “off.”

Radio buttons, however, operate as a group and provide mutually exclusive selection values. A user can select only one option in a radio button group.]

11. 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.

12. 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 en- ableEvents() method is used by objects that handle events by overriding their eventdis- patch methods.

13. 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

14. What is the use of JList?

JList is a Swing component with which we can display a list of elements. This component also allows the user to select one or more elements visually.

15. 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 Compo- nent.

16. 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.

17. 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, select- ing a list, etc. There are two types of models for handling events and they are: a) event- inheritance model and b) event-delegation model

18. Which containers use a border Layout as their default layout?

The window, Frame and Dialog classes use a border layout as their default layout.

19. What are the differences between AWT and Swing?

SWING	AWT
Swing components are light weight.	AWT components are heavy weight.
Swing components are drawn by java itself that's why it'splatform independent.	AWT components are platform dependent.
Look and Feel of swing can be changed.	There is no such feature in AWT.
All of the buttons, entry fields, etc. are drawn by the Swing package on the drawing surface provided by the window object. This is the reason that Swing has more code.	AWT is a thin layer of code on top of OS.
Swing components are generally slower than AWT.	Use of native peers speeds component performance.
Swing supports a wider range of features like icons and pop-up tool-tips for components.	AWT components do not support features like icons and tool-tips.
Swing has many advanced features like JTabel, Jtabbed pane which is not available in AWT.	These feature is not available in AWT.

20. What is the use of WindowListener?

WindowListener interface is implemented in class to handle following activi- ties

Opening a window	- Showing a window for the first time
Closing a window	- Removing the window from the screen
Iconifying a window	- Reducing the window to an icon on the desktop.
Deiconifying a window	- Restoring the window to its original size.
Focused window	- The window which contains the "focus owner".
Activated window (frame or dialog)	- This window is either the focused window, or owns the focused window.
Deactivated window	- This window has lost the focus. For more information about focus, see the AWT Focus Subsystem specifica- tion.
Maximizing the window	- Increasing a window's size to the maximum allowable size, either in the vertical direction, the horizontal direc- tion, or both directions.

21. Difference between AWT and Swing

There are many differences between java awt and swing that are given below.

Java AWT	Java Swing
AWT components are platform-dependent.	Java swing components are platform-independent.
AWT components are heavyweight.	Swing components are lightweight.
AWT doesn't support pluggable look and feel.	Swing supports pluggable look and feel.
AWT provides less components than Swing.	Swing provides more powerful componentssuch as tables, lists, scrollpanes, colorchooser, tabbedpane etc.
AWT doesn't follow MVC (Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.	Swing follows MVC.

22. What is a container class?

Container classes are classes that can have other components on it. So for creating a GUI, we need at least one container object. There are 3 types of containers.

Panel: It is a pure container and is not a window in itself. The sole purpose of a Panel is to organize the components on to a window.

Frame: It is a fully functioning window with its title and icons.

Dialog: It can be thought of like a pop-up window that pops out when a message has to be displayed. It is not a fully functioning window like the Frame.

23. 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.

24. 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 dif- ferent sizes. The default Layout Manager of Panel and Panel sub classes is FlowLayout.

25. Why would you use Swing Utilities.invoke And Wait or Swing Utilities. invoke Later?

To make update in a Swing component but not in a callback. And If the update to be hap- pened immediately (perhaps for a progress bar component) then use invokeAndWait. Whenthe update response is not need immediately, then use invokeLater.

26. 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, select- ing a list, etc. There are two types of models for handling events and they are: a) event- inheritance model and b) event-delegation model.

27. 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.

28. 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 Sys- tem.exit to terminate the JVM.

29. 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 Compo- nent.

30. 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

31. 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 en- ableEvents() method is used by objects that handle events by overriding their eventdis- patch methods.

32. 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 hierar- chy.

33. 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.

34. 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.

35. 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.

PART-B

1. Create a simple menu application that enables a user to select one of the following items: (16) (IT2301 APR/MAY-2015)

   a. Radio1 b. Radio2 c. Radio3 d. Radio4 e. Radio5

   i. From the menu bar of the application

   ii. From a pop-up menu

   iii. From a toolbar

2. Add tooltips to each menu item that indicates some information about the Radio station such as type of music and its broadcast frequency.

3. Write a program to create a frame with the following menus, such that the corresponding geometric object is created when a menu is clicked. (IT2301 MAY/JUNE-2014)

   a. Circle. (4)

   b. Rectangle. (4)

   c. Line. (4)

   d. Diagonal for the rectangle. (4)

4. Write a program to stimulate the layout and working of a calculator. (16) (IT2301 MAY/ JUNE-2014)

   Explain in detail about AWT event hierarchy. (16) (IT2301 MAY/JUNE-2020