An event can be defined as changing the state of an object or behavior by performing actions. Actions can be a button click, cursor movement etc.

Event handling is a mechanism to control the events and to decide what should happen after an event.

It has sources and listeners.

→ Events are generated from the sovrce such as buttons, checkboxes, text-components etc.

→ Listners are used for ~~f~~ handling the events generated from the source, listeners represent interfaces that are responsible for handling events.

The super keyword in java is a reference variable that is used to refer to the immediate parent class.

→ Super is used to call a superclass method.

→ Super is used to call a superclass constructor

→ Super must be the first statement in a constructor.

→ Cannot be used in a static context.

Final keyword in java is used as a non-access modifier used to restrict a user.

(1) If you make variable as final, you cannot change it's value.

(2) If you make method as final, you cannot override it.

(3) If you make class as final, you cannot extend it.

The static keyword in java means that a variable or a method is shared between all instances of that class. It is mainly used for memory management

1) Static variable is same for all the instances of a class.

2) Static method can be invoked without creating an instance of a class.

→ This, super cannot be used in a static context.

→ Static block is used to initialise static data member.
It is executed before the main method

→ A class can be made static only if it is nested

Dynamic method dispatch is the mechanism in which a call to an overridden method is resolved at runtime instead of compile time.

Upcasting is a technique in which a superclass reference variable refers to the object of subclass.

~~Allows Java~~
Runtime polymorphism or dynamic method dispatch ~~to define~~ allows super class to define as well as share it's own method and also allows the subclasses to define their own implementation.

toString() is a method which belongs to the Object class. whenever we try to print the reference of the object toString() method is automatically invoked and prints Class name @ and the hexadecimal representation of objects hashcode.

Exception can be recovered by using the try-catch block.	An error cannot be recovered.
Classified into two checked or unchecked	Are always unchecked
belongs to java.lang.Exception package	java.lang.Error package

Checked Exceptions occur at compile time and unchecked at run time.
example of unchecked exceptions
nullpointerException, ArrayIndexoutofBounds etc

An Interface is a blueprint of a class which contains methods that are ~~only~~ all abstract.
Interface cannot be instantiated.

used for abstraction and multiple inheritance

A class cannot extend an interface but it can implement interface. And the class which implements a particular interface has to provide the definitions of methods declared in interface

The access modifiers in java speeify the accessibility of a class, method, field or a constructor.

1) private → only within its own class.
2) Default → within the package
3) protected → within the package and outside the package through child class.
4) public → everywhere.

Non access modifiers → final, static, abstract provide the info about the characteristics of class, method, or variable.

The java command line argument is an argument passed at the time of running the java program.
The arguments passed from the console can be recieved in the java program and can be used as input.

A method with variable length arguments (varargs) can have zero or multiple arguments.
Varargs are most useful when number of arguments is unknown. They also reduce the code as overloaded methods are not required

e.g: public void (String... str)
{ }.

Checked

These are the exceptions that are checked at compile time. If some code within a method throws a Checked exception, then the method must either handle the exception or it must Specify the exception using throws keyword.

IOException, SQLException, FileNotFoundException. ClassNotFoundException.

Unchecked.

These are the exceptions that are not Checked at compile time but during runtime. Consider the following java program. It compiles fine, but it throws ArithmeticException when run. The compiler allows it to run because it is an unchecked exception.

ArrayIndexOutOfBoundsException, NullPointerException, ArithmeticException.

(1) try block contains a set of statements where an exception can occur.

(2) catch block is used to handle the uncertain condition of a try block.

(3) throw keyword is used to explicitly throw an exception. we can throw both checked and unchecked exceptions.

(4) throws keyword indicates that a method might throw one of the listed type exceptions. The caller to these methods has to handle the exception using a try-catch block.

5) Java finally block is always executed whether an exception occurs or not.

finilize is the method in java which is used to perform clean up processing just before object is garbage collected. Garbage collector calls it before destroying any object.

Inner class or nested class is a class that is declared inside the class or interface.

```
Class Outer_class
 { //code
   Class Inner_class{
   //code
         }
   }
```

Advantages

1) It can access all the members (data members and methods) of outer class, including private.
2) used to develop more readable and maintainable code.
3) code optimization: it requires less code to write.

Inner class can be Non-static as well as static.

A static class is a class that is created inside a class, is called a static nested class in java. It can be accessed by outer class name.

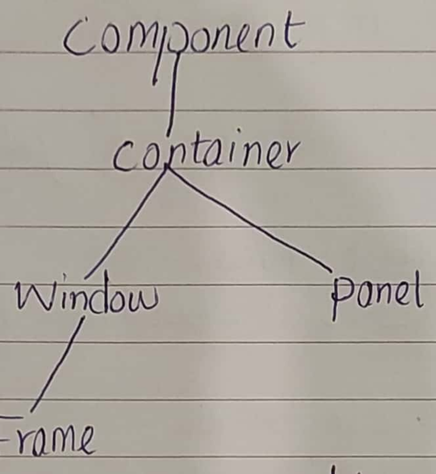
(1) It can access static data members of the outer class (including private).
(2) Static nested class cannot access non-static data members.

AWT	SWING
It is an API used to develop window-based applications in java	Swing is a Graphical user interface and a part of ~~Java~~ Java foundation classes that are used to design different applications.
It's Components are heavy weighted.	It's components are Light weighted.
AWT is platform dependent.	Swing is platform independent.
It has Less functionality Compared to Swing	It has more functionality Compared to AWT.
requires more time for execution.	requires Less time for execution.

Container is a Subclass of component in AWT that can Contain another Components like buttons, textfields, labels. It is basically a Screen where components are placed at their specific locations.

panel is a ~~Simplest~~ Subclass of container class. It provides Space in which an application can attach any other component. It inherits the Container class.

Frame is the container that contain the title bar and border and can have menu bars. It can have other components like buttons, textfields etc. Frame is most widely used container ~~while~~ while developing an AWT application.



At the top of the hierarchy is the component class which is an abstract class that encapsulates all of the attributes of a visual component.

Swing is built on AWT

Swing Component are derived from JComponent class.
JComponent inherits the AWT classes Container and component.

JFrame does not inherit JComponent but it does inherit ~~&~~ AWT Component and Container.

JPanel however inherits JComponent.

Steps for GUI Creation

1 Import required packages
e.g: java.awt.*;
jarax.Swing.*;

2 Setup the top level container.
JFrame frame = new JFrame();

3 Get component area of the top level container.
Container c = frame.getContentPane();

4 Apply Layout to that Area.
c.SetLayout(new FlowLayout());

5 Create and add components
JButton b1 = new JButton("Hello");
c.add(b1);

6 Set the size of Frame and make it visible
frame.SetSize(200,200);
frame.SetVisible(true);

flowlayout components arrange themselves from left to right in the order they were added

Gridlayout components arrange themselves in a matrix formation (rows, columns)

Borderlayout is used to arrange components in five regions north, South, east, west and center.

JDBC steps.

(1) import JDBC packages.

e.g:
```
import java.sql.*;
```

(2) Register the Driver class.

```
class.forName("Com.mysql.jdbc.Driver");
```

(3) Create a connection

```
String url = "jdbc:mysql://localhost:3306/database";
String Username = "User";
String password = "password";

Connection connection = DriverManager.getConnection(
                          url, username, password);
```

(4) Create a Statement

```
Statement statement = Connection.createStatement();
```

(5) Execute SQL Queries.

```
ResultSet resultset = Statement.executeQuery("
                 SELECT * FROM mytable");
```

(6) Close connection

```
resultset.close();
Statement.close();
Connection.close();
```

What is JDBC?

JDBC Stands for Java Database Connectivity is a Standard Java API for database-independent connectivity between the Java Programming language and a wide range of databases.

e.g:

1) Making a connection to database
2) Creating SQL or MySQL statements etc.