**Purpose of Layout managers, why are there different Layout managers, some of the features of individual Layout managers (such as difference between BorderPane vs GridPane:**

The LayoutManagers are used to arrange components in a particular manner. LayoutManager is an interface that is implemented by all the classes of layout managers.

Layout nodes using Pane, StackPane, FlowPane, GridPane, BorderPane, HBox, and VBox:

The **stackpane** will attempt to resize each child to fill its content area. If the child could not be sized to fill the stackpane (either because it was not resizable or its max size prevented it) then it will be aligned within the area using the alignment property, which defaults to Pos.CENTER.

A **GridPane** arranges nodes in a grid (matrix) formation. The nodes are placed in the specified column and row indices.

A **BorderPane** can place nodes in five regions: top, bottom, left, right, and center, using the setTop(node), setBottom(node), setLeft(node), setRight(node), and setCenter(node) methods.

**HBox** places the nodes in one row.

**VBox** places the nodes in one column.

**What are advantages of using**

**Should exception handling be used for all programming situations:**

Exception handling enables a program to deal with exceptional situations and continue its normal execution.

RuntimeException, Error, and their subclasses are known as **unchecked** exceptions. All other exceptions are known as **checked** exceptions, meaning that the compiler forces the programmer to check and deal with them in a try-catch block or declare it in the method header.

There are 5 keywords used in java exception handling: **try; catch; finally; throw; throws**

Java try block is used to enclose the code that might throw an exception. It must be used within the method.

Java try block **must be followed** by either **catch or finally** block.

If you didn’t catch one exception, using Finally you’ll guarantee to catch a specific line code.

Exception handling: because sometimes we can solve the problem, instead of catch the exception

Ex: constructor with negative numbers: new Circle(-10); It’s better catch the exception to avoid that problem

**What are event handlers what must be done to use an event handler:**

An event handler processes the event fired from the source object. See page 586

Or: Changing the state of an object is known as an event. For example, click on button, dragging mouse etc.

Event handler: 2 steps 🡪 1: list for event 2: register it

**Binary I/O** does not involve encoding or decoding and thus is more efficient than text I/O.

**Text I/O:** we can save data. Reading and writing in a file.

Objects be written to and read from files:

Yes, they must be in binary. This process is called serialization.

After a serialized object has been written into a file, it can be read from the file and deserialized that is, the type information and bytes that represent the object and its data can be used to recreate the object in memory.

Computers do not differentiate between binary files and text files. All files are stored in binary format, and thus all files are essentially binary files.

ComboBox allows single choices

ListView allows multiple choices

CheckBoxes selects multiple option

RadioButtons solects only one option (Toggle group)

TextFiled – single row

TextArea – multiple rows