

CISC124 – Today's Topics		
<ul style="list-style-type: none"> <li>▪ Quiz 3 topics</li> <li>▪ GUI applications - summary <ul style="list-style-type: none"> <li>▪ Event driven programming</li> <li>▪ Implementation architectures (MVC)</li> </ul> </li> <li>▪ Java packages <ul style="list-style-type: none"> <li>▪ AWT model</li> <li>▪ Swing model</li> <li>▪ JavaFX model</li> </ul> </li> <li>▪ JavaFX examples</li> </ul>		
Winter 2019	F. de la Parra	1

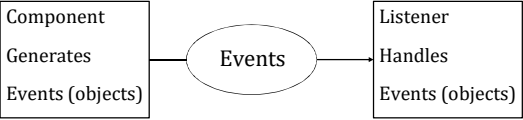
•1

Quiz 3 Schedule		
<p>Location and Hours:</p> <ul style="list-style-type: none"> <li>• Quiz 3 will be written in Jeffery 155, at the beginning of your lab session</li> <li>• Mon Mar 25, 9:30 am – 10:30 am</li> <li>• Mon Mar 25, 2:30 pm – 3:30 pm</li> <li>• Tue Mar 26, 8:30 am – 9:30 am</li> <li>• Wed Mar 27, 2:30 pm – 3:30 pm</li> </ul> <p>Format:</p> <ul style="list-style-type: none"> <li>• One coding question</li> <li>• A set of multiple choice questions</li> </ul>		
Winter 2019	F. de la Parra	2

•2

Quiz 3 Topics		
<p>Topics (Everything covered from Feb 15 until Mar 20) :</p> <ul style="list-style-type: none"> <li>• Number systems (decimal, hexadecimal, octal, binary). Base changes</li> <li>• Internal representation of numbers (two's complement and IEEE754)</li> <li>• Complements. Binary addition and subtraction. Round-off error</li> <li>• Javadoc</li> <li>• Aspects of testing</li> <li>• Junit framework</li> <li>• Class inheritance</li> <li>• Abstract, anonymous and inner classes</li> <li>• Interfaces</li> <li>• Generics. ArrayList&lt;T&gt; class</li> <li>• Lambda expressions</li> </ul>		
Winter 2019	F. de la Parra	3

•3

Event Driven Programming		
 <pre> graph LR     subgraph ComponentBox [Component]         direction TB         C1[Generates]         C2[Events (objects)]     end     subgraph ListenerBox [Listener]         direction TB         L1[Handles]         L2[Events (objects)]     end     Events((Events))     ComponentBox --&gt; Events     Events --&gt; ListenerBox </pre> <ul style="list-style-type: none"> <li>▪ GUI programming paradigm: <ul style="list-style-type: none"> <li>▪ Application (potentially) never ends until the user stops it (endless loop)</li> <li>▪ Program execution flow is mostly determined by the user. Programmer only offers possible alternatives of execution</li> <li>▪ Event generating components have specific predetermined behaviours</li> <li>▪ Event Listeners need to understand these behaviours and respond to events</li> <li>▪ Event Listeners need to implement the functionality specified to meet the requirements and purpose of the application</li> </ul> </li> </ul>		
Winter 2019	F. de la Parra	4

•4

GUI Architectural Patterns		
<ul style="list-style-type: none"> <li>▪ MVC (Model – View – Controller) <ul style="list-style-type: none"> <li>▪ Model → Processes user input, updates the state of the application, retrieves any required data, updates the View and communicates its state to the controller → Business logic</li> <li>▪ View → Displays data to the user in a predetermined layout</li> <li>▪ Control → Handles user input and forwards it to the Model according to the application's state</li> </ul> </li> <li>▪ Others <ul style="list-style-type: none"> <li>▪ Hierarchical MVC: Implements GUI application as a hierarchy of Controllers. Each Controller is associated with its own View – Model pair</li> <li>▪ PAC (Presentation – Abstraction – Control): Implements GUI application as a hierarchy of cooperating agents. Higher level agents implement business logic. Each agent has PAC pattern</li> </ul> </li> </ul>		
Winter 2019	F. de la Parra	5

•5

Java Packages		
<ul style="list-style-type: none"> <li>▪ AWT (Abstract Windowing Toolkit) <ul style="list-style-type: none"> <li>▪ Older package (considered obsolete). Three categories of classes</li> <li>▪ Graphics → These classes define colours, fonts, images, polygons, etc.</li> <li>▪ Components → Visible GUI components: buttons, menus, dialog boxes, etc.</li> <li>▪ Layout Managers → Control the layout of components within container objects</li> </ul> </li> <li>▪ Swing <ul style="list-style-type: none"> <li>▪ Evolution of the AWT package to support the MVC architectural model</li> <li>▪ Classes focus more on managing the data of an application than only on its presentation aspects</li> <li>▪ More sophisticated components</li> </ul> </li> </ul>		
Winter 2019	F. de la Parra	6

•6

Java Packages		
<ul style="list-style-type: none"> <li>▪ JavaFX                             <ul style="list-style-type: none"> <li>▪ Common programming model for different platforms (desktop, web, mobile)</li> <li>▪ Richer and more advanced set of components</li> <li>▪ Integration of advanced graphics, 3D graphics support</li> </ul> </li> </ul>		
Winter 2019	F. de la Parra	7

•7