C.1.1

What is an event, and how are events generated? What does it mean to register for an event?

An event is any action that can occur in a computer. Clicking on a mouse, receiving a socket connection, or pressing a key are examples of events. Events are generated when components trigger the events. To "register for an event" means to associate an event listener with an event, for example a mouse listener.

C.1.3

What is a `Container`? A `Panel`? The `contentPane`?

A container is a data storage structure designed to hold any object type. In the context of AWT and Swing, the container holds graphical objects (Components — buttons, labels, etc.). A panel is a container to hold and display buttons, etc. on a Java frame. The content pane is a container within a JFrame in which non-menu items should be added.

C.1.5

Explain the effect of executing each of the following statements. What can you day about class `HelloMonitor`?

```
JButton aButton = new JButton("Hello");
aButton.addActionListener(new HelloMonitor());
panel.add(aButton);
// Put the panel into the frame.
getContentPane().add(panel);
```

```
JButton aButton = new JButton("Hello"); //creates a new button with the caption "Hello."
aButton.addActionListener(new HelloMonitor()); //Adds an action listener to the button
panel.add(aButton); //Adds the button to a panel
// Put the panel into the frame
getContentPane().add(panel); //Adds the panel to a content pane
```

Class HelloMonitor implements the ActionListener interface.

C.2.1

_____ is the top-level container for GUI applications.

JFrame is the top-level container for GUI applications.

C.2.3

A `JPanel` can also be used to _____.

A `JPanel` can also be used to draw shapes and images.

C.3.1

Describe the characteristics of the following layout managers:
Border Layout
Box Layout
Grid Layout
Flow Layout

BorderLayout: Arranges objects in five areas of the container. These areas are North, West, Center, East, and South.
BoxLayout: Arranges all objects in a single row or column.
GridLayout: Arranges the objects in a two-dimensional grid.
FlowLayout: Arranges objects in a left-to-right order across the container, flowing down to the next row when a row is filled.

C.3.3

Why are layout managers useful?

The layout managers provides a way to position the components within a display area without having to manipulate graphic coordinates.

C.4.1

Which of the following data entry components (check box, radio button, combo box, or text field) is appropriate for the following situations?

a. Recording answers in a multiple-choice exam
b. Entering the state or province in a data-entry form that is collecting address information
c. Entering the city in a data entry form that is collecting address information
d. Allow the user to select several different breakfast items from a menu

a. Radio button
b. Combo box
c. Text field
d. Check box

C.5.1

How can you obtain formatting with three decimal places?

There are two possible approaches:
One is to use the NumberFormat class as follows:

```
numberForm = NumberFormat.getNumberInstance();
numberForm.setMaximumFractionDigits(3);
numberForm.setMinimumFractionDigits(3);
numberForm.format(doubleValue);
```

And the other is to use the String.format method (which calls on the Formater class) as follows:
`String.format("%.3f", doubleValue)`

C.6.1

Why must we extend the abstract class `Drawable` to include the methods `getName`, `getMnemonic`, `getSortcut` and `getIcon`?

We extend the `DrawableInt` *interface to include the methods* `getName`*, getMnemonic, getShortcut, and getIcon so that we can use a DrawableInt as a data field in the DrawableIcon class that can be implemented as any shape. The methods must then be implemented in all of the classes implementing the DrawableInt interface, therefore implementing it for the DrawableIcon class.*