

## AWT

- 1) The various controls supported by AWT are
  - a. Labels, push buttons
  - b. Checkboxes, choice, list
  - c. Scroll bars, text area, text field
  - d. **All of these**
  
- 2) The concept of the menu bar can be implemented by using three Java classes—
  - a. MenuBar
  - b. Menu
  - c. MenuItem
  - d. **All of these**
  
- 3) The most commonly used layout managers are
  - a. FlowLayout
  - b. BorderLayout
  - c. GridLayout
  - d. CardLayout
  - e. **All of these**
  
- 4) The constructor which the Text Event class defines.
  - a. **TextEvent(Object source, int event\_type)**
  - b. textevent (Object source, int event\_type)
  - c. textevent (object Source, float event\_type)
  - d. textevent (Object source, string event\_type)
  
- 5) In Java an event is an \_\_\_\_\_ which specifies the change of state in the source.
  - a. Class
  - b. **Object**
  - c. Int
  - d. String
  
- 6) The name of the event classes are
  - a. ActionEvent, ComponentEvent
  - b. ContainerEvent, FocusEvent
  - c. ItemEvent, KeyEvent
  - d. WindowListener, MouseEvent
  - e. TextEvent
  - f. **All of these**
  
- 7) The classes and interfaces defined in AWT are contained within the \_\_\_\_\_ package.
  - a. **java.awt.\***
  - b. java.sql.\*
  - c. java.io.\*
  - d. java.int\*
  
- 8) Java packages such as \_\_\_\_\_ support the Event handling mechanism.
  - a. java.util
  - b. java.awt

- c. java.awt.event
- d. **All of these**

- 9) The general form to set a specific type of layout manager is
- a. **void setLayout(LayoutManager lm)**
  - b. Void setLayout(LayoutManager lm)
  - c. void setLayout(layoutManager lm)
  - d. Void setLayout(Layoutmanager lm)

- 10) Some of the event listener interfaces are\_\_\_\_\_
- a. ActionListener, ComponentListener
  - b. ContainerListener, FocusListener
  - c. ItemListener, KeyListener
  - d. WindowListener, MouseListener
  - e. TextListener
  - f. **All of these**

- 11) The AWT container is an instance of the \_\_\_\_\_ class which holds various components and other containers
- a. Graphics
  - b. **Container**
  - c. Eventobj
  - d. None of these

- 12) A checkbox is a control that consists of a
- a. Combination of a small box
  - b. A label
  - c. Combination of a large box and a label
  - d. **Both a & b**

- 13) Java applets are used to create \_\_\_\_\_ applications
- a. Graphical
  - b. User interactive
  - c. **Both a & b**
  - d. None of these

- 14) In Java, events are all the activities that occur between
- a. The user
  - b. The applications
  - c. **Both a & b**
  - d. None of these
  - e.

- 15) AWT means
- a. **Abstract Window Toolkit**
  - b. Abstract Window Toollayout
  - c. Abstract Withdraw Tools
  - d. Abstract Window Title

- 16) Till now, two models have been introduced in Java for
- a. **Receiving and Processing events**

- b. Deleting and Processing events
  - c. Receiving and Deleting events
  - d. Modifying and Receiving events
- 17) An event is generated when the internal state of the event source is \_\_\_\_\_
- a. Not changed
  - b. **Changed**
  - c. Either changed or not
  - d. None of these
- 18) EventObject class belongs to:
- a. **java.util**
  - b. java.awt
  - c. java.lang
  - d. java.sql
- 19) Positions the components into five regions: east, west, north, south, center
- a. **BorderLayout**
  - b. CardLayout
  - c. GridLayout
  - d. FlowLayout
- 20) Arranges the components as a deck of cards such that only one component is visible at a time
- a. BorderLayout
  - b. **CardLayout**
  - c. GridLayout
  - d. FlowLayout
- 21) Arranges the components horizontally
- a. BorderLayout
  - b. CardLayout
  - c. GridLayout
  - d. **FlowLayout**
- 22) Arranges the components into grid
- a. BorderLayout
  - b. CardLayout
  - c. **GridLayout**
  - d. FlowLayout
- 23) \_\_\_\_\_ creates a dropdown list of textual entries
- a. **Choice**
  - b. Checkbox
  - c. Textbox
  - d. TextComponent
- 24) The Component class and MenuComponent class are the \_\_\_\_\_ which represent the GUI components.
- a. Subclasses
  - b. **Superclasses**

- c. Both a & b
- d. None of these

25) The Component class is an abstract class and so its \_\_\_\_\_ are used to create components.

- a. **Subclasses**
- b. Superclasses
- c. Both a & b
- d. None of these

e.

26) The AWT classes can be roughly categorized into the following groups:

- a. GUI Components
- b. Layouts
- c. Graphics Tools
- d. Event Handlers
- e. **All of these**

27) Panel is used for \_\_\_\_\_ components

- a. **Grouping**
- b. Managing
- c. Deleting
- d. Modifying

28) An Applet is a \_\_\_\_\_ of Panel:

- a. **Subclass**
- b. Superclass
- c. Both a & b
- d. None of these

29) Window is used for \_\_\_\_\_ windows

- a. Creating
- b. Handling
- c. Modifying
- d. **Both a & b**

30) The subclasses of Window are

- a. Dialog
- b. Frame
- c. **Both a & b**
- d. None of these

e.

31) The CardLayout class defines the following constructors:

- a. **CardLayout() // First**  
**CardLayout(int hor, int ver) //second**
- b. **CardLayout() // First**  
**CardLayout(int hour, int ver) //second**

- c. `CardLayout()` // First  
`Cardlayout(int hor, int var)` //second
- d. `CardLayout()` // First  
`Cardlayout(int hour, int ver)` //second

32) A menu bar represents

- a. **A list of menus which can be added to the top of a top-level window**
- b. A list of menus which can be deleted to the top of a top-level window
- c. A list of menus which can be added to the bottom of a bottom-level window
- d. None of these

33) Each menu is associated with a \_\_\_\_\_ list of menu items:

- a. Checkbox
- b. **Drop-down**
- c. Choice
- d. None of these

34) The two types of menus which are given as follows:

- a. Pop-up menu
- b. Regular menu
- c. **Both a & b**
- d. None of these

35) Regular menus are placed at the \_\_\_\_\_ of the application window within a menu bar

- a. **Top**
- b. Bottom
- c. Top-down
- d. Botttom-up

36) The \_\_\_\_\_ interface is used to handle the menu events

- a. ContainerListener
- b. FocusListener
- c. **ActionListener**
- d. WindowListener
- e.

37) The text field and text area controls create a \_\_\_\_\_ area respectively

- a. Single-line text
- b. Multi-line text
- c. **Both a & b**
- d. None of these

e.

38) The \_\_\_\_\_ interface handles list events:

- a. ContainerListener
- b. FocusListener
- c. **ActionListener**

d. **ItemListener**

39) The \_\_\_\_\_ interface handles choice events:

- a. ContainerListener
- b. **ItemListener**
- c. ActionListener
- d. WindowListener

40) The \_\_\_\_\_ interface is used to handle checkbox events:

- a. ContainerListener
- b. **ItemListener**
- c. ActionListener
- d. WindowListener

41) The \_\_\_\_\_ interface is used to handle button events:

- a. ContainerListener
- b. ItemListener
- c. **ActionListener**
- d. WindowListener

42) A push button is an active control that has a \_\_\_\_\_ appearance

- a. One dimensional
- b. Two dimensional
- c. **Three dimensional**
- d. None of these

43) ItemListener is defined by the \_\_\_\_\_ method:

- a. itemStateChanged()
- b. **itemStateChanged()**
- c. itemChanged()
- d. itemState()

44) \_\_\_\_\_ is a superclass of TextField and TextArea classes that is used to create single-line or multiline textfields respectively:

- a. TextBox
- b. CheckBox
- c. **TextComponent**
- d. Choice

e.

45) A label is a simple control which is used to display \_\_\_\_\_ on the window:

- a. **Text(non-editable)**
- b. Text(editable)
- c. Both a & b
- d. None of these

46) The Delegation Event Model defines a \_\_\_\_\_ approach to handle events:

- a. **Logical**
- b. Physical
- c. Both a & b

d. None of these

47) The Delegation Event Model is based on the concept of \_\_\_\_\_

- a. Source
- b. Listener
- c. **Both a & b**
- d. None of these

48) A source generates an event and sends it to \_\_\_\_\_ listeners that can handle the event:

- a. One
- b. Two
- c. **One or more**
- d. None of these

49) Applet is a small program written in the Java programming language that performs:

- a. A specific task
- b. May or may not be a part of a larger program
- c. **Both a & b**
- d. None of these

50) \_\_\_\_\_ generates action events when an item is double-clicked, generate action events when an item is selected or deselected.

- a. **List**
- b. Check box
- c. Menu item
- d. Text box