Practical No 05 - Manual Answer

Advance Java Programming (Government Polytechnic, Pune)

Assignment No 05

Page No 26

ix. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specifications	Quantity	Remarks (if any)
1	Computer System	Lenovo Desktop, Intel Core i3, 2.40 GHz, 4.00 GB RAM	30 (Batch Size)	For all practicals which are Window-based applications
2	Operating System	Windows 8.0, 32-bit OS, x64-based processor		
3	Development Software	JDK 1.8.0_181, Command Prompt, Editors - NetBeans or Eclipse		

Page No 26 and 27

x. Program Code

1. Write a program which creates Menu of different colors and disable menu item for Black color.

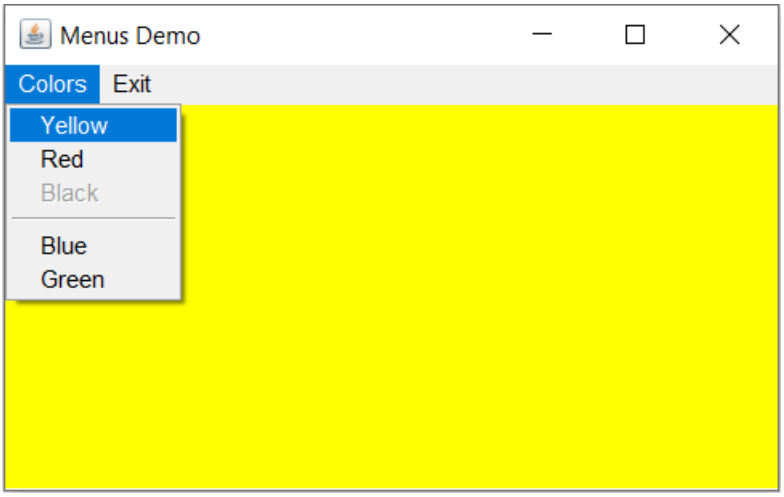
```
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Menu;
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ColorsMenuDemo extends Frame implements ActionListener
{
	MenuBar mainMenuBar;
	Menu mColors, mExit;
	MenuItem miYellow, miRed, miBlue, miGreen, miBlack, miClose;

	public ColorsMenuDemo() {
		this.setTitle("Menus Demo");
		this.setSize(400, 250);
		this.setLayout(new FlowLayout());

		this.mainMenuBar = new MenuBar();
		this.setMenuBar(this.mainMenuBar);
```

```
        this.mColors = new Menu("Colors");
        this.miYellow = new MenuItem("Yellow");
        this.miYellow.addActionListener(this);
        this.mColors.add(this.miYellow);

        this.miRed = new MenuItem("Red");
        this.miRed.addActionListener(this);
        this.mColors.add(this.miRed);

        this.miBlack = new MenuItem("Black");
        this.miBlack.addActionListener(this);
        this.mColors.add(this.miBlack);
        this.miBlack.setEnabled(false);

        this.mColors.addSeparator();

        this.miBlue = new MenuItem("Blue");
        this.miBlue.addActionListener(this);
        this.mColors.add(this.miBlue);

        this.miGreen = new MenuItem("Green");
        this.miGreen.addActionListener(this);
        this.mColors.add(this.miGreen);

        this.mainMenuBar.add(this.mColors);

        this.mExit = new Menu("Exit");
        this.miClose = new MenuItem("Close");
        this.miClose.addActionListener(this);
        this.mExit.add(this.miClose);

        this.mainMenuBar.add(this.mExit);

        this.setVisible(true);
    }

    public static void main(String[] args) {
        new ColorsMenuDemo();
    }

    @Override
```

```
        public void actionPerformed(ActionEvent e) {
            if(e.getSource() == this.miYellow) {
                this.setBackground(Color.YELLOW);
            } else if(e.getSource() == this.miRed) {
                this.setBackground(Color.RED);
            } else if(e.getSource() == this.miBlack) {
                this.setBackground(Color.BLACK);
            } else if(e.getSource() == this.miBlue) {
                this.setBackground(Color.BLUE);
            } else if(e.getSource() == this.miGreen) {
                this.setBackground(Color.GREEN);
            } else if(e.getSource() == this.miClose) {
                System.exit(0);
            }
        }
}
```

Page No 27

xi. Result (Output of Code)



Page No 27 and 28

xii. Practical Related Questions

1. Write the use of setEnabled() method.

The menu or menu item can be enabled or disable by using setEnable() method.

2. Write the procedure to assign shortcut key to the MenuItem.

A menu object is created with the Menu class. The menus can be accessed via keyboard as well. To bind a menu to a particular key, we use the **`setMnemonic()`** method.

In following example, the menu can be opened with the 'Alt + S' shortcut.

```
MenuItem miSave = new MenuItem("Save");
miSave.setMnemonic(KeyEvent.VK_S);
```

The other way to add shortcut is to create the object of **MenuShortcut** Class and later add to MenuItem in its constructor, as shown in following example –

```
MenuShortcut ms = new MenuShortcut(KeyEvent.VK_B);
MenuItem miBlack = new MenuItem("Black", ms);
```

3. Write a syntax and use of addSeparator() method.
The addSeparator() method is used to add a line between Menu items added to Menu, and it acts as separator. The syntax of method is –

```
menuObject.addSeparator();
```

Page No 28 and 29

XIII. Exercise

1. Find errors in following program and display output as shown below.
(Refer to the program given in manual.)

Corrected Program

```
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Menu;
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.MenuShortcut;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;

public class MenusDemo extends Frame implements ActionListener {

	MenuBar mainMenuBar;
	Menu mFile;
	MenuItem miNew, miOpen, miSaveAs, miExit;
	MenuShortcut msExit;

	public MenusDemo() {
		this.setTitle("Menus Demo");
		this.setSize(400, 250);
```

```
        this.setLayout(new FlowLayout());

        this.mainMenuBar = new MenuBar();
        this.setMenuBar(this.mainMenuBar);

        this.mFile = new Menu("File");
        this.mainMenuBar.add(this.mFile);

        this.miNew = new MenuItem("New ...");
        // this.miNew.addActionListener(this);
        this.mFile.add(this.miNew);

        this.miOpen = new MenuItem("Open ...");
        // this.miOpen.addActionListener(this);
        this.mFile.add(this.miOpen);

        this.miSaveAs = new MenuItem("Save As ...");
        // this.miSaveAs.addActionListener(this);
        this.mFile.add(this.miSaveAs);

        this.mFile.addSeparator();

        this.msExit = new MenuShortcut(KeyEvent.VK_X);
        this.miExit = new MenuItem("Exit", this.msExit);
        this.miExit.addActionListener(this);
        this.mFile.add(this.miExit);

        this.setVisible(true);
    }

    public static void main(String[] args) {
        new MenusDemo();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == this.miExit) {
            System.exit(0);
        }
    }
}
```