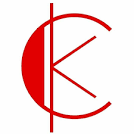
**KC GROUP OF RESEARCH & PROFESSIONAL INSTITUTE PANDOGA UNA H.P**

****

**CORE JAVA LAB**

**Subject Code:- CS-512**

**Department of Computer Science & Engineering**

**Submitted By : YUVRAJ TIWARI Submitted To : ER. PRIYANKA**

**Affiliated to**

**HIMACHAL PRADESH TECHNICAL UNIVERSITY,HAMIRPUR**

|  |  |
| --- | --- |
| **Sr.No.** | **PRACTICAL** |
| 1 | Demonstrating the use of method of math class. |
| 2 | Programs to implement the methods of string class. |
| 3 | To demonstrate the interface. |
| 4 | To demonstrate inheritance. |
| 5 | To demonstrate super and this. |
| 6 | To demonstrate static variable and methods. |
| 7 | To demonstrate exceptions |
| 8 | To demonstrate file input stream and file output stream classes |
| 9 | To demonstrate mouse and keyboard events in an applet |
| 10 | To demonstrate the creation of a frame. |
| 11 | To demonstrate checkboxes with proper events |
| 12 | To demonstrate scroll bars with proper events |
| 13 | To demonstrate menu bars and menus |
| 14 | To demonstrate dialog boxes |

**Program – 1**

**Demonstrating the use of methods in math class.**

**package** infoJava;

**public** **class** Opr {

**public** **static** **void** main(String[]args)

{

**int** a=10;

**int** b=20;

**double** c=2.145;

System.***out***.println(Math.*max*(a, b));

System.***out***.println(Math.*min*(a, b));

System.***out***.println(Math.*abs*(c));

System.***out***.println(Math.*cos*(c));

System.***out***.println(Math.*multiplyExact*(a,b));

}

}

**Output:**

**20**

**10**

**2.145**

**-0.5431663470050839**

**20**

**Program – 2**

**Programs to implement the methods of string class.**

package infoJava;

public class swintut {

public static void main(String[] args) {

String s="Yuvraj";

System.out.println(s.length());

System.out.println(s.charAt(1));

System.out.println(s.replace('a', 'b'));

System.out.println(s.startsWith("pr"));

System.out.println(s.toLowerCase());

System.out.println(s.endsWith("ka"));

System.out.println(s.substring(2, 5));

System.out.println(s.equals("Yuvraj"));

}

}

**Output:**

8

r

Yuvraj

true

Yuvraj

true

iya

true

**Program - 3**

**To demonstrate interface**

importjava.io.\*;

*// A simple interface*

interfaceIn1

{

*// public, static and final*

finalinta = 10;

*// public and abstract*

voiddisplay();

}

// A class that implements the interface.

classTestClass implementsIn1

{

*// Implementing the capabilities of*

*// interface.*

publicvoiddisplay()

{

System.out.println("interface Example");

}

publicstaticvoidmain (String[] args)

{

TestClass t = newTestClass();

t.display();

System.out.println(a);

}

}

**Output:**

interface Example

10

**Program - 4**

**To demonstrate inheritance.**

1. **class** Employee{

2. **float** salary=40000;

3. }

4. **class** Programmer **extends** Employee{

5. **int** bonus=10000;

6. **public static void** main(String args[]){

7. Programmer p=**new** Programmer();

8. System.out.println("Programmer salary is:"+p.salary);

9. System.out.println("Bonus of Programmer is:"+p.bonus);

10. }

11. }

**Output:**

Programmer salary is:40000.0

Bonus of programmer is:10000

**Program -5**

**To demonstrate super and this.**

1. **class** Animal{

2. String color="white";

3. }

4. **class** Dog **extends** Animal{

5. String color="black";

6. **void** printColor(){

7. System.out.println(color);//prints color of Dog class

8. System.out.println(**super**.color);//prints color of Animal class

9. }

10. }

11. **class** TestSuper1{

12. **public static void** main(String args[]){

13. Dog d=**new** Dog();

14. d.printColor();

15. }}

**Output:**

black

White

**Program – 6**

**To demonstrate static variable and methods.**

1. **class** Calculate{

2. **static int** cube(**int** x){

3. **return** x\*x\*x;

4. }

5.

6. **public static void** main(String args[]){

7. **int** result=Calculate.cube(5);

8. System.out.println(result);

9. }

10. }

**Output:**

**125**

**Program - 7**

**To demonstrate exceptions**

1. public class JavaExceptionExample{

2. public static void main(String args[]){

3. try{

4. //code that may raise exception

5. int data=100/0;

6. }catch(ArithmeticException e){System.out.println(e);}

7. //rest code of the program

8. System.out.println("rest of the code...");

9. }

10. }

**Output**

Exception in thread main java.lang.ArithmeticException:/ by zero rest of the code...

**Program – 8**

**To demonstrate file input stream and file output stream classes**

**File Input stream**

1. **import** java.io.FileOutputStream;

2. **public class** FileOutputStreamExample {

3. **public static void** main(String args[]){

4. **try**{

5. FileOutputStream fout=**new** FileOutputStream("D:\\testout.txt"); 6. fout.write(65);

7. fout.close();

8. System.out.println("success...");

9. }**catch**(Exception e){System.out.println(e);} } }

**Output**

Success...

**File Output Stream**

1. **import** java.io.FileOutputStream;

2. **public class** FileOutputStreamExample {

3. **public static void** main(String args[]){

4. **try**{

5. FileOutputStream fout=**new** FileOutputStream("D:\\testout.txt"); 6. String s="Welcome to javaTpoint.";

7. **byte** b[]=s.getBytes();//converting string into byte array 8. fout.write(b);

9. fout.close();

10. System.out.println("success...");

11. }**catch**(Exception e){System.out.println(e);} }

**Output** Success...

**Program – 9**

**To demonstrate mouse and keyboard events in an applet**

1. importjava.applet.\*;

2. importjava.awt.\*;

3. importjava.awt.event.\*;

4.

5. /\* <APPLET CODE="MouseEventsDemo.class" WIDTH="300" HEIGHT="300"> 6. </APPLET>

7. \*/

8.

9. publicclassMouseEventsDemoextendsAppletimplementsMouseListener,Mou seMotionListener{

10.

11. String message ="";

12.

13. publicvoidinit(){

14.

15. setBackground(Color.YELLOW);

16. addMouseListener(this);

17. addMouseMotionListener(this);

18. }

19.

20. publicvoidpaint(Graphics g){

21. g.drawString(message,50,50);

22. }

23.

24. publicvoidmouseEntered(MouseEvent me){

25. setBackground(Color.PINK);

26. message ="Mouse Entered: ("+me.getX()+", "+me.getY()+")"; 27. repaint();

28. }

29.

30. publicvoidmouseExited(MouseEvent me){

31. setBackground(Color.RED);

32. message ="Mouse Exited: ("+me.getX()+", "+me.getY()+")";

33. repaint();

34. }

35.

36. publicvoidmouseClicked(MouseEvent me){

37. setBackground(Color.CYAN);

38. message ="Mouse Clicked: ("+me.getX()+", "+me.getY()+")"; 39. repaint();

40. }

41.

42. publicvoidmousePressed(MouseEvent me){

43. setBackground(Color.MAGENTA);

44. message ="Mouse Pressed: ("+me.getX()+", "+me.getY()+")"; 45. repaint();

46. }

47.

48. publicvoidmouseReleased(MouseEvent me){

49. setBackground(Color.GREEN);

50. message ="Mouse Released: ("+me.getX()+", "+me.getY()+")"; 51. repaint();

52. }

53.

54. publicvoidmouseMoved(MouseEvent me){

55. setBackground(Color.ORANGE);

56. message ="Mouse Moved: ("+me.getX()+", "+me.getY()+")"; 57. repaint();

58. }

59.

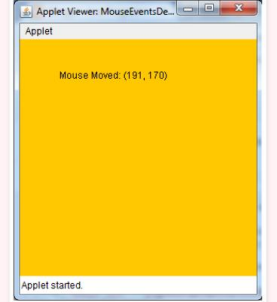
60. publicvoidmouseDragged(MouseEvent me){

61. setBackground(Color.GRAY);

62. message ="Mouse Dragged: ("+me.getX()+", "+me.getY()+")"; 63. repaint();

64. }

**Output**

****

**Program – 10**

**To demonstrate the creation of a frame.**

import javax.swing.\*;

// inheriting JFrame

public class test2 extends JFrame

{

JFrame frame;

test2()

{

setTitle("this is also a title");

// create button

JButton button = new JButton("click");

button.setBounds(165, 135, 115, 55);

// adding button on frame

add(button);

// setting close operation

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 500);

setLayout(null);

setVisible(true);

}

public static void main(String[] args)

{

new test2();

}

}

**Output :**

****

**Program – 11**

**To demonstrate checkboxes with proper events**

1. **import** java.awt.\*;

2. **public class** CheckboxExample

3. {

4. CheckboxExample(){

5. Frame f= **new** Frame("Checkbox Example");

6. Checkbox checkbox1 = **new** Checkbox("C++");

7. checkbox1.setBounds(100,100, 50,50);

8. Checkbox checkbox2 = **new** Checkbox("Java", **true**);

9. checkbox2.setBounds(100,150, 50,50);

10. f.add(checkbox1);

11. f.add(checkbox2);

12. f.setSize(400,400);

13. f.setLayout(**null**);

14. f.setVisible(**true**);

15. }

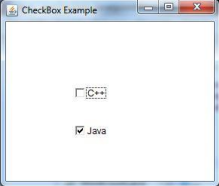
16. **public static void** main(String args[])

17. {

18. **new** CheckboxExample();

19. }

**Output:**

****

**Program – 12**

**To demonstrate scroll bars with proper events**

1. **import** java.awt.\*;

2. **import** java.awt.event.\*;

3. **class** ScrollbarExample{

4. ScrollbarExample(){

5. Frame f= **new** Frame("Scrollbar Example");

6. **final** Label label = **new** Label();

7. label.setAlignment(Label.CENTER);

8. label.setSize(400,100);

9. **final** Scrollbar s=**new** Scrollbar();

10. s.setBounds(100,100, 50,100);

11. f.add(s);f.add(label);

12. f.setSize(400,400);

13. f.setLayout(**null**);

14. f.setVisible(**true**);

15. s.addAdjustmentListener(**new** AdjustmentListener() { 16. **public void** adjustmentValueChanged(AdjustmentEvent e) { 17. label.setText("Vertical Scrollbar value is:"+ s.getValue()); 18. }

19. });

20. }

21. **public static void** main(String args[]){

22. **new** ScrollbarExample();

23. }

24. }

**Output:**

****

**Program – 13**

**To demonstrate menu bars and menus**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.\*;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.scene.control.\*;

import javafx.stage.Stage;

import javafx.scene.control.Alert.AlertType;

import java.time.LocalDate;

public class MenuBar\_1 extends Application {

// launch the application

public void start(Stage s)

{

// set title for the stage

s.setTitle("creating MenuBar");

// create a menu

Menu m = new Menu("Menu");

// create menuitems

MenuItem m1 = new MenuItem("menu item 1");

MenuItem m2 = new MenuItem("menu item 2");

MenuItem m3 = new MenuItem("menu item 3");

// add menu items to menu

m.getItems().add(m1);

m.getItems().add(m2);

m.getItems().add(m3);

// create a menubar

MenuBar mb = new MenuBar();

// add menu to menubar

mb.getMenus().add(m);

// create a VBox

VBoxvb = new VBox(mb);

// create a scene

Scene sc = new Scene(vb, 500, 300);

Practical File 18BT110316

// set the scene

s.setScene(sc);

s.show();

}

public static void main(String args[])

{

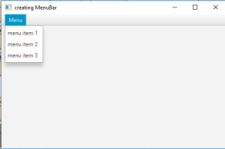
// launch the application

launch(args);

}

}

**Output**:



**Program – 14**

**To demonstrate dialog boxes**

1. **import** java.awt.\*;

2. **import** java.awt.event.\*;

3. **public class** DialogExample {

4. **private static** Dialog d;

5. DialogExample() {

6. Frame f= **new** Frame();

7. d = **new** Dialog(f , "Dialog Example", **true**);

8. d.setLayout( **new** FlowLayout() );

9. Button b = **new** Button ("OK");

10. b.addActionListener ( **new** ActionListener()

11. {

12. **public void** actionPerformed( ActionEvent e )

13. {

14. DialogExample.d.setVisible(**false**);

15. }

16. });

17. d.add( **new** Label ("Click button to continue."));

18. d.add(b);

19. d.setSize(300,300);

20. d.setVisible(**true**);

21. }

22. **public static void** main(String args[])

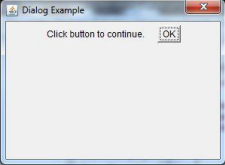
23. {

24. **new** DialogExample();

25. }

26. }

**Output:**

****