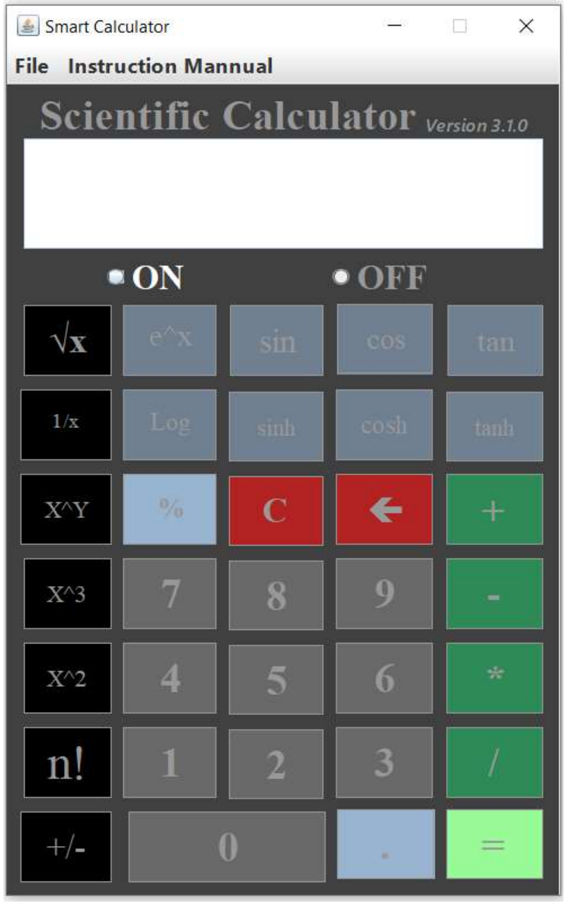
Smart Calculator



Developed By

MANI KUMAR ADAPALA

Aim : To Build a smart calculator using java to make our Daily calculations and conversions easy.

Explanation Of The Code :

1. A Jframe is created in required size.
2. A Label named “SCIENTIFIC CALCULATOR” was created and affixed as Title.
3. Next, we created a Main Input console for the user as a JTextField to take input from the user and to display output After computation.
4. Then we created two Radio Buttons using JRadioButton as ON and OFF.
5. We made both of them as a ButtonGroup so that only one button works at a time either on/off.
6. We created all necessary Buttons using JButtons and named them as we required.
7. Now the basic interface is ready. Now it’s time for Back coding part for the result computation.
8. For that we set different operators like “ + - * / % ^ ” .
9. If “+” is selected, we take two inputs and add them and display the result as output on the JTextField.
10. If “-” is selected, we take two inputs and subtract them and display the result as output on the JTextField.
11. If “*” is selected, we take two inputs and multiply them and display the result as output on the JTextField.
12. If “/” is selected, we take two inputs and divide them and display quotient as output on the JTextField.
13. If “%” is selected, we take two inputs and divide them and display the remainder as output on the JTextField.
14. If “^” is selected, we take two inputs and raise power accordingly to them and display the result as output on the JTextField.
15. If “n^2” is selected, we take input and compute the square of that number and display the result as output on the JTextField.
16. If “n!” is selected, we take input and compute the factorial of that number and display the result as output on the JTextField.
17. If “e^x” is selected we take input and compute the exponent of that number using math.exponential() and display the result as output on the JTextField.
18. If “log” is selected we take input and compute the Logarithm of that number using math.log() and display the result as output on the JTextField.
19. We also added some trignometric and hyperbolic trignometric functions like Sine, Cosine, Tan, Sinh, Cosh, Tanh.

20. We compute those values with math.sin() , math.cos(), math.tan(), math.sinh(), math.cosh() and math.tanh().
21. We added a clear button which clears all input and output on JTextField with TextField.setText(null) .
22. For the backspace button we converted input into a string buffer and removed the last indexed value from that and we displayed bacon TextField.
23. We have done the back code for all the buttons with " ActionListener " and all the actions were performed according to their selections.
24. Now, It's time to design the Number converter.
25. For that we chose JCheckBox. Four checkboxes were created as Decimal, Octal, Binary and Hex and again those all are grouped as a ButtonGroup so that only one checkbox works at a time.
26. Now, Four TextFields are created to each checkbox for input and output purposes.
27. We again took care that only the TextField against the selectedcheckBox only enabled and remaining were disabled. With the help of textField.setenabled(true) and textField.setenabled(false).
28. Now if any of them were selected we take input from the corresponding textField only. Then a JButton called convert was created which converts one integer form to all.
29. We will convert on the principle called Integer.parseInt(number,base); and Integer.toBinaryString(), Integer.toOctalString(), Integer.toHexString().
30. A JButton called clear was created and it's action was set to clear everything in the Number converter TextFields. This can be Achieved By TextField.setText(null).
31. Now Our Number Converter is ready !! It's Time to Create an Ascii Converter.
32. For this we created two JLabels called Input Data and Ascii Value.
33. Two TextFields were created to get input and to display Ascii Output.
34. A JButton called Convert was created and it's action was set to convert given data into American Standard Code for Information Interchange.
35. That can be achieved by, str.getBytes("US-ASCII");
36. Here We need to catch the " UnsupportedEncodingException. "
37. And FInal ASCII Output will be printed as an array with textPane.setText(Arrays.toString(bytes)).
38. A JButton called clear Ascii was created and it's action was set to clear everything in the Ascii converter TextFields. This can be Achieved By TextField.setText(null).
39. Finally We need to add a Menu Bar for an Application look.
40. We can add that with JMenu and we need to add menu items as JMenuItems.
41. New Menu Bar called File was added. Inside it, we added three menu Items called "Standard Calculator" , "Number Converter" , "Ascii Converter".

42. If we click on any of them it will extend the JFrame up to that.
43. We also Added Keyboard Shortcuts for user convenience. like, mntmNewMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, InputEvent.SHIFT_MASK));
44. Now for User clarity we introduced a new menu item called Instruction Manual. If clicked it will automatically find the user's system’s default browser or app to view the pdf and shows the Instruction Manual.
45. Foe this we created a folder called “FILE” in the same directory as our java code and added Manual.pdf to it.
46. We used Desktop.getDesktop().open(new java.io.File("File\\Mannual.pdf"));
47. Here we need to catch the exceptions and try to show the file.
48. Now, the main thing comes, that Everything in the calculator must disabled if OFF and should be enabled on pressing ON.
49. To achieve this set this state ment to all, btn/TextField/Radiobutton.setenabeled(false) for OFF and btn/TextField/Radiobutton.setenabeled(true) for ON.
50. That’s It we all Done Our Calculator APP is Ready !

⟵ THE END ⟶