Assignment 3 :: Event Handling & Graphics

Readme

1. You have to send me a zip/tar file containing all your source code. The filename of this archive file **MUST** be something like p123456.zip or p123456.tar.gz or p123456.tar.bz etc. etc., i.e., your roll-number.
2. Inside the archive file, I only require a single file per code/question. The filename must be p123456-code-01.java, p123456-code-02.java, p123456-code-03.java, and so on.
3. If you want to add additional data related to a question, please feel free to use names like p123456-code-01.jpg, p123456-code-01.txt, and so on.
4. DO NOT include your source code in microsoft word documents. Take care of indentation and formatting.

Code 1: Differentiating between Event Sources

Write code in which you have two buttons titled “Button 1” and “Button 2”. Associate event listeners with both buttons. When user clicks on the first button, the screen should read “Button 1 was pressed”. When user clicks on the second button, the screen should read “Button 2 was pressed”. The button identification must be extracted from the event given to the handler function.

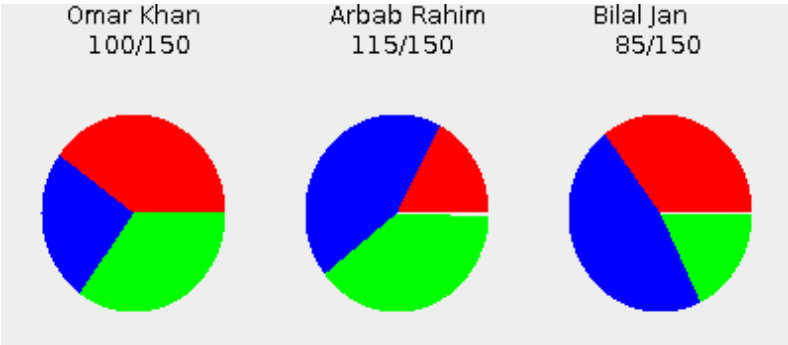
Code 2: Grabbing Mouse Cursor Location

Write a Java code comprising of a simple blank window. When the mouse moves within the window, the pointer position should be dumped on terminal.

Code 3: Grabbing Mouse Cursor Location (Adapter Version)

Convert Code 1 such that you use Adapters instead of Listeners

Code 4: Marks Pie-Chart



Write code in which you have 3 arrays containing names of students, their marks in English, Urdu, Maths, etc. Example is below:

```
String name[] = { "Omar Khan", "Arbab Rahim", "Bilal Jan" };
int    engl[] = { 40, 20, 30 };
int    urdu[] = { 25, 50, 40 };
int    math[] = { 35, 45, 15  };
```

Write code for the paintComponents() method where you draw pie-charts for the students as pictured above. The total marks are 150. Each pie should represent proportionate marks obtained. (See slide 56 for drawing arcs).

Code 5: Mouse Drawing

Write code in which you have a canvas on which you can draw stuff using your mouse. The mouse movement will be captured by drawing a series of small lines using **drawLine(startx, starty, endx, endy)**. For this to work, you can capture the **startx** and **starty** in **public void mousePressed()**. The **endx** and **endy** can be captured in **public void mouseDragged()**. Please note that after drawing a small line segment in mouseDragged(), you need to continue updating the **startx** and **starty** in the same function.

Code 6: Moving drawString()

Write code in which you write your name using a drawString(). Then, using your numeric keypad (top, left, right, bottom buttons), move this text around the screen. You can use the following steps for guidance:

1. Add a keyListener() to a JPanel within your code
2. Overwrite the keyPressed() handler
3. Check for the key pressed using ev.getKeyCode()
4. If the keycode is equal to KeyEvent.VK_DOWN, adjust variable y value accordingly.
5. If the keycode is equal to KeyEvent.VK_UP, adjust the y value accordingly.
6. Apply similar behaviour for KeyEvent.VK_LEFT, and KeyEvent.VK_RIGHT
7. Draw the string using drawString() with the adjusted x and y positions above

Code 7: Multi-Line drawString()

Extend the Code 6 above by extending the code to support multiple lines. For this, you may apply the following additional steps:

1. Set the font of the canvas to anything of your choosing
2. Obtain the FontMetrics for the font which you have applied to the canvas
3. If the width of the string to type is less than the window width, just use drawstring() to write the complete string
4. Else, tokenize the string into components. For example, if you have a String s, then the tokenization will follow:
   String[] words = s.split(“”);
   Run a loop across the words array and check for each word (or combination of words). If the width is less than the window width, use drawString() to draw the partial string. For the remaining, adjust the y position from the fontMetrics information. Run until the words array is exhausted.