|  |  |
| --- | --- |
| *A close up of a logo  Description automatically generated* | *DEPARTMENT OF COMPUTER ENGINEERING* |

|  |  |
| --- | --- |
| Semester | S.E. Semester III – Computer Engineering |
| Subject | Object Oriented Programming Using Java (Skill Based Lab) |
| Subject Professor In-charge | Prof. Indu Anoop |
| Laboratory | Online Lab |

|  |  |  |
| --- | --- | --- |
| Student Name | Trisha Shah | |
| Roll Number | 20102A0004 | |
| Grade and Subject Teacher’s Signature |  |  |

|  |  |  |
| --- | --- | --- |
| Experiment | 15 | |
| Problem Statement | Program on Graphics Class | |
| Resources / Apparatus Required | Hardware: Computer System | Software: jdk 1.8, Eclipse / Notepad++/IntelliJ IDEA |
| Details | The Graphics class is the abstract base class for all graphics contexts that allow an application to draw onto components that are realized on various devices, as well as onto off-screen images.  A Graphics object encapsulates state information needed for the basic rendering operations that Java supports. This state information includes the following properties:  The Component object on which to draw.  A translation origin for rendering and clipping coordinates.  The current clip.  The current color.  The current font.  The current logical pixel operation function (XOR or Paint).  The current XOR alternation color (see setXORMode(java.awt.Color)).  Coordinates are infinitely thin and lie between the pixels of the output device. Operations that draw the outline of a figure operate by traversing an infinitely thin path between pixels with a pixel-sized pen that hangs down and to the right of the anchor point on the path. Operations that fill a figure operate by filling the interior of that infinitely thin path. Operations that render horizontal text render the ascending portion of character glyphs entirely above the baseline coordinate. | |
| Code | import java.awt.\*;  import java.applet.\*;  public class applet4 extends Applet {  public void paint(Graphics g) {  Color c1=new Color(255,0,0);  g.setColor(c1);  g.drawLine(20,10,20,50);  g.fillRect(80,10,80,50);  Color c2=new Color(0,0,255);  g.setColor(c2);  g.fillOval(20,110,100,80);  }  } | |
| Output |  | |
| Conclusion | Thus, we could successfully implement the graphics class. | |