Michael Lenehan - Assignment 7

Question 1

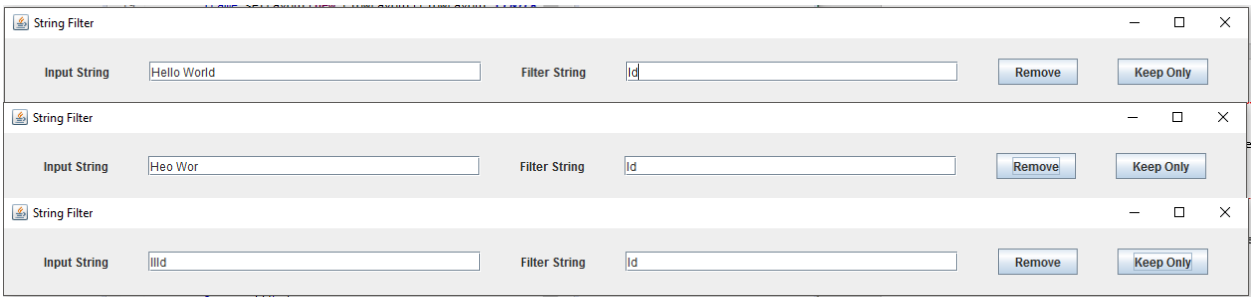
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class UpperLower implements ActionListener{

	protected String in;
	JFrame frame = new JFrame("UpperLower");
	JTextField input = new JTextField(30);
	JButton button = new JButton("Change Case");

	public UpperLower()
	{
		frame.setLayout(new FlowLayout(FlowLayout.CENTER, 40, 20));

		frame.add(input);
		input.addActionListener(this);

		frame.add(button);
		button.addActionListener(this);

		frame.pack();
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}

	public void actionPerformed(ActionEvent e)
	{
		if(e.getActionCommand().equals("Change Case"))
		{
			String in = input.getText();
			input.setText(ChangeCase(in));
			System.out.println(ChangeCase(in));
		}
	}

	public static String ChangeCase(String str)
	{
		char[] c = str.toCharArray();
		if(str.length() == 0)
			return "Enter a String";
		else
		{
			for(int i = 0; i<str.length()-1; i++)
			{
				if(Character.isLowerCase(str.charAt(i)))
				{
					c[i] = Character.toUpperCase(str.charAt(i));
				}
```

```
                        else c[i] = Character.toLowerCase(str.charAt(i));
                }
        }
        return str = String.valueOf(c);
    }


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

Question 2

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Filter implements ActionListener{

    protected String in, fil;
    protected int i, j;
    JFrame frame = new JFrame("String Filter");
    JLabel l1 = new JLabel("Input String");
    JTextField input = new JTextField(30);
    JLabel l2 = new JLabel("Filter String");
    JTextField filter = new JTextField(30);
    JButton b1 = new JButton("Remove");
    JButton b2 = new JButton("Keep Only");

    public Filter()
    {
        frame.setLayout(new FlowLayout(FlowLayout.CENTER, 40, 20));
        frame.add(l1);
        frame.add(input);
        input.addActionListener(this);
        frame.add(l2);
        frame.add(filter);
        filter.addActionListener(this);
        frame.add(b1);
```

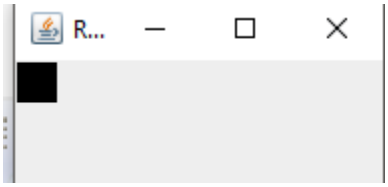
```
            b1.addActionListener(this);
            frame.add(b2);
            b2.addActionListener(this);
            frame.pack();
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setVisible(true);
      }

      public void actionPerformed(ActionEvent e)
      {
            String in = input.getText(), fil = filter.getText();
            if(e.getActionCommand().equals("Remove"))
            {
                  if(in.length() == 0) input.setText("Enter a String");
                  else
                  {
                        input.setText(in.replaceAll("[" +fil + "]+", ""));
                  }
            }
            if(e.getActionCommand().equals("Keep Only"))
            {
                  if(in.length()==0) input.setText("Enter a String");
                  else
                  {
                        input.setText(in.replaceAll("[^" +fil + "]+", ""));
                  }
            }
      }

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



Question 3

```
import java.awt.*;
import javax.swing.*;

public class RectangleComponent extends JComponent{
      private int width, height;
      private Color colour;
```

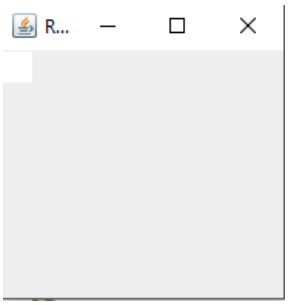
```
	public RectangleComponent(int w, int h, Color c)
	{
		width = w;
		height = h;
		colour = c;
	}

	public void paintComponent(Graphics g)
	{
		g.setColor(colour);
		g.fillRect(1, 1, width, height);
	}

	public static void main(String args[])
	{
		JFrame frame = new JFrame("Rectangle");
		frame.add(new RectangleComponent(20, 20, Color.BLACK));
		frame.setSize(200, 100);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
}
```



Question 4

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class RectangleComponent extends JComponent implements MouseListener{
	private int width, height;
	private Color colour;

	public RectangleComponent(int w, int h, Color c)
	{
		width = w;
		height = h;
		colour = c;
		this.addMouseListener(this);
	}

	public void paintComponent(Graphics g)
	{
		g.setColor(colour);
		g.fillRect(1, 1, width, height);
	}
```

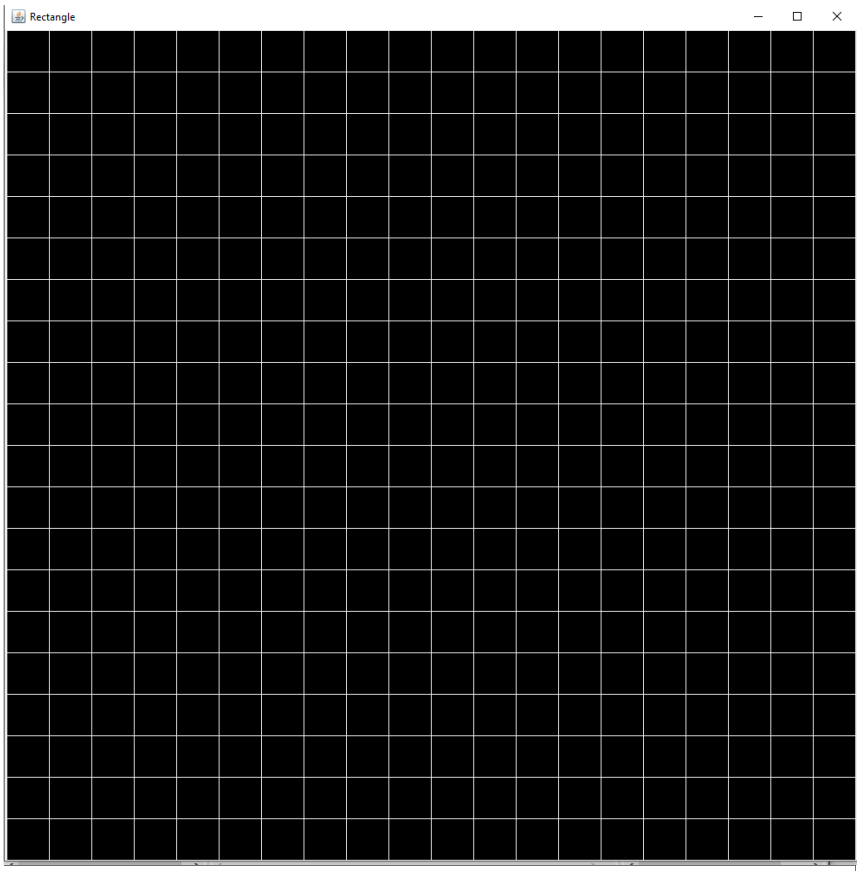
```
	public void mouseClicked(MouseEvent e)
	{
		if(colour == Color.BLACK)
		{
			colour = Color.WHITE;
			repaint();
		}
		else
		{
			colour = Color.BLACK;
			repaint();
		}

	}

	public void mouseEntered(MouseEvent e){}
	public void mouseExited(MouseEvent e){}
	public void mousePressed(MouseEvent e){}
	public void mouseReleased(MouseEvent e){}

	public static void main(String args[])
	{
		JFrame frame = new JFrame("Rectangle");
		RectangleComponent rect = new RectangleComponent(20, 20, Color.BLACK);
		frame.add(rect);
		frame.setSize(200, 200);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
}
```

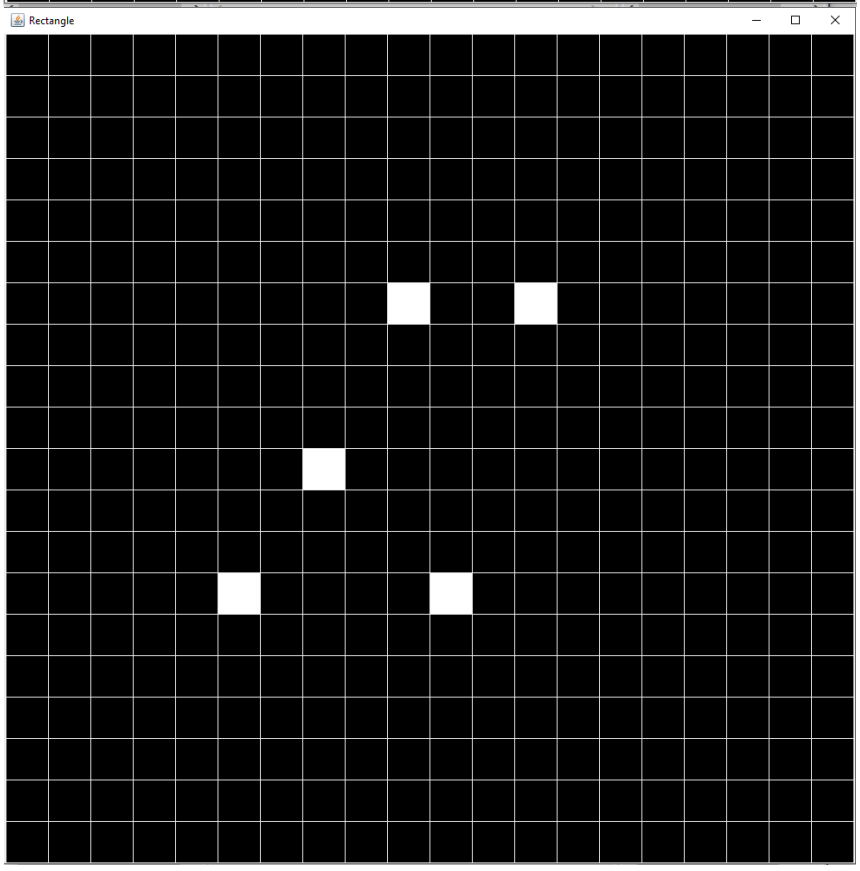


Question 5

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class RectangleComponent extends JComponent implements MouseListener{
	private int width, height;
	private Color colour;

	public RectangleComponent(int w, int h, Color c)
	{
		width = w;
```

```
		height = h;
		colour = c;
		this.addMouseListener(this);
	}

	public void paintComponent(Graphics g)
	{
		for(int i = 0; i<20; i++)
		{
			for(int j = 0; j<20; j++)
			{

				g.setColor(colour);
				g.fillRect(1, 1, width, height);
			}
		}
	}

	public void mouseClicked(MouseEvent e)
	{
		if(colour == Color.BLACK)
		{
			colour = Color.WHITE;
			repaint();
		}
		else
		{
			colour = Color.BLACK;
			repaint();
		}

	}

	public void mouseEntered(MouseEvent e){}
	public void mouseExited(MouseEvent e){}
	public void mousePressed(MouseEvent e){}
	public void mouseReleased(MouseEvent e){}

	public static void main(String args[])
	{
		JFrame frame = new JFrame("Rectangle");
		frame.setLayout(new GridLayout(20, 20));
		for(int i = 0; i<20*20; i++)
		{
			RectangleComponent rect = new RectangleComponent(200, 200,
Color.BLACK);
			frame.add(rect);
		}
		frame.setSize(1000, 1000);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
}
```

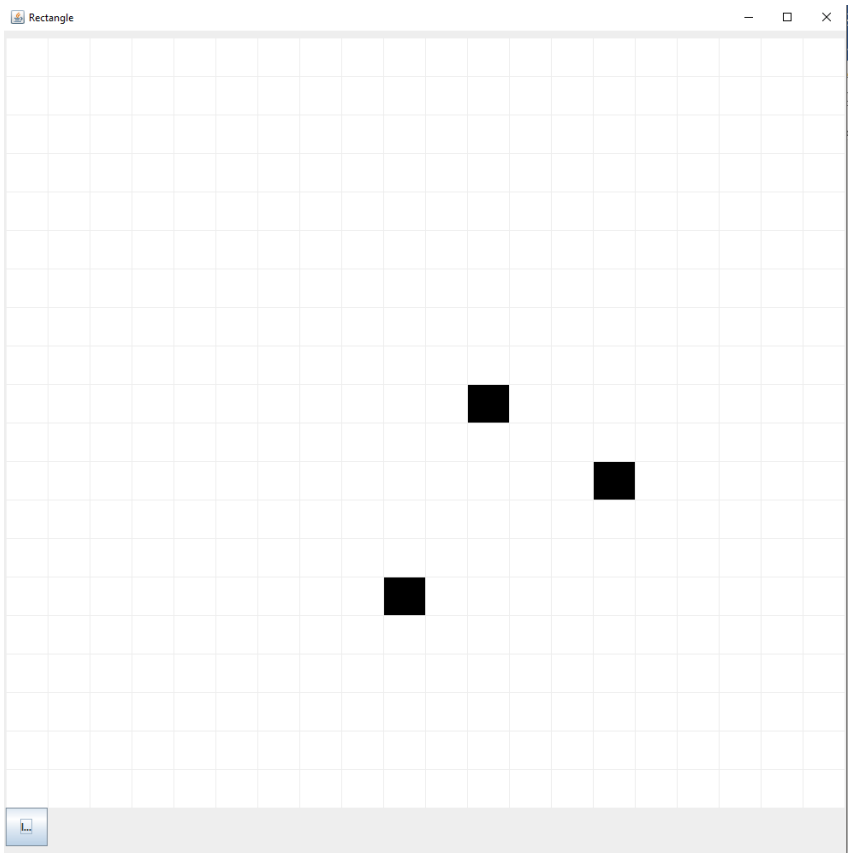

Rectangle

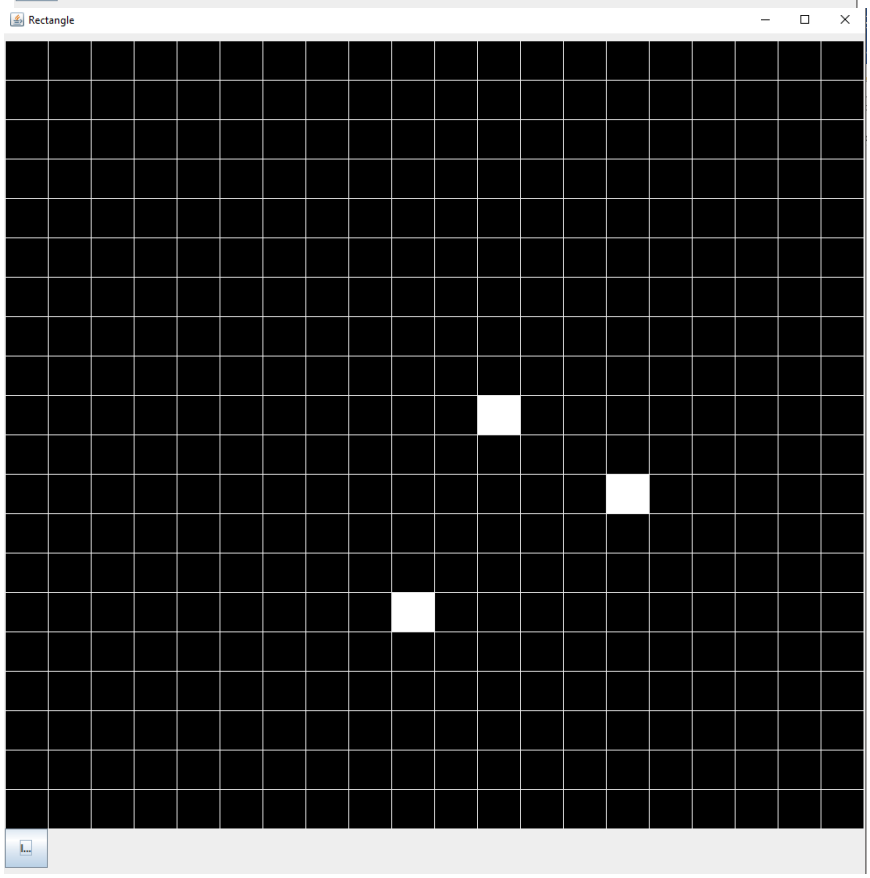

Rectangle

Question 6

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class RectangleComponent extends JComponent implements MouseListener,
ActionListener{

	private int width, height;
	private Color colour;
	static JFrame frame = new JFrame("Rectangle");
	protected static JButton b = new JButton("Invert");

	public RectangleComponent(int w, int h, Color c) // Rectangle constructor
	{
		width = w;
		height = h;
		colour = c;
		this.addMouseListener(this);
	}

	public void paintComponent(Graphics g) // Paints all 400 Rectangles, and adds
buttons action listener
	{
		for(int i = 0; i<20; i++)
		{
			for(int j = 0; j<20; j++)
			{

				g.setColor(colour);
				g.fillRect(1, 1, width, height);
			}
		}
		b.addActionListener((ActionListener) this);
	}

	public void mouseClicked(MouseEvent e) //Gives colour changes on mouse clicks
	{
		if(colour == Color.BLACK)
		{
			colour = Color.WHITE;
			repaint();
		}
		else
		{
			colour = Color.BLACK;
			repaint();
		}
	}

	public void mouseEntered(MouseEvent e){}
	public void mouseExited(MouseEvent e){}
	public void mousePressed(MouseEvent e){}
```

```
	public void mouseReleased(MouseEvent e){}

	public void actionPerformed(ActionEvent e) //Method reacts slowly if some
squares have already been changed
                                                                        //Changes
Colour of all Rectangles in the frame
	{
		if(e.getActionCommand().equals("Invert"))
		{
			if(colour == Color.BLACK)
			{
				colour = Color.WHITE;
				repaint();
			}
			else
			{
				colour = Color.BLACK;
				repaint();
			}
		}
	}

	public static void main(String args[])
	{
		frame.setLayout(new GridLayout(21, 20));
		for(int i = 0; i<20*20; i++)
		{
			RectangleComponent rect = new RectangleComponent(200, 200,
Color.BLACK);
			frame.add(rect);
		}
		b.setSize(new Dimension(200, 600));
		frame.add(b);
		frame.setSize(1000, 1000);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
}
```

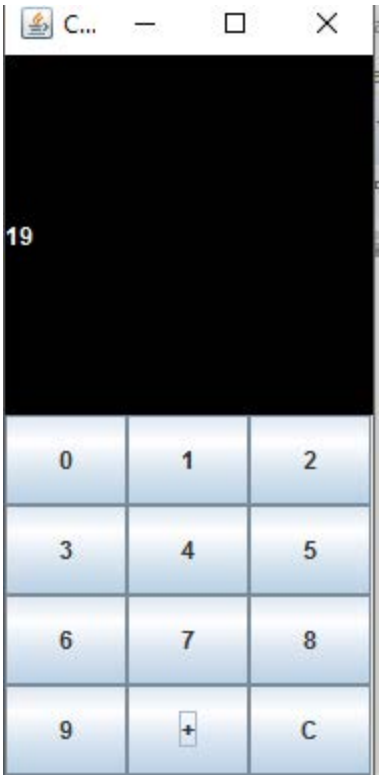

Rectangle


Rectangle

Question 7

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Calculator implements ActionListener{
	protected int c = 0, first_num = 0, val = 0, new_val;
	JFrame frame = new JFrame("Calculator");
	JPanel topPan = new JPanel(), botPan = new JPanel();
	JLabel screen = new JLabel();
	boolean bool = false;

	public Calculator()
	{
		frame.setSize(200, 400);
		frame.setLayout(new GridLayout(2, 1));
		frame.add(topPan);
		topPan.setSize(200, 100);
		frame.add(botPan);
		topPan.setLayout(new BorderLayout(20, 20));
		topPan.add(screen, BorderLayout.CENTER);
		screen.setOpaque(true);
		screen.setBackground(Color.BLACK);
		screen.setForeground(Color.WHITE);
		botPan.setLayout(new GridLayout(4, 3));
		for(int i = 0; i<12; i++)
		{
			JButton[] b = new JButton[12];
			if(i<10)
			{
				b[i] = new JButton("" + i + "");
				botPan.add(b[i]);
				b[i].addActionListener(this);
			}
			else
			{
				if(i == 10)
				{
					b[i] = new JButton("+");
					botPan.add(b[i]);
					b[i].addActionListener(this);
				}
				else
				{
					b[i] = new JButton("C");
					botPan.add(b[i]);
					b[i].addActionListener(this);
				}
			}
		}
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
```

```
	public void actionPerformed(ActionEvent e)
	{
		if(c<10)
		{
			if(bool == true)
			{
				screen.setText("");
				bool = false;
			}
			switch(e.getActionCommand().toString())
			{
			case "0": {screen.setText(screen.getText().concat("0")); c++;};
break;
			case "1": {screen.setText(screen.getText().concat("1")); c++;};
break;
			case "2": {screen.setText(screen.getText().concat("2")); c++;};
break;
			case "3": {screen.setText(screen.getText().concat("3")); c++;};
break;
			case "4": {screen.setText(screen.getText().concat("4")); c++;};
break;
			case "5": {screen.setText(screen.getText().concat("5")); c++;};
break;
			case "6": {screen.setText(screen.getText().concat("6")); c++;};
break;
			case "7": {screen.setText(screen.getText().concat("7")); c++;};
break;
			case "8": {screen.setText(screen.getText().concat("8")); c++;};
break;
			case "9": {screen.setText(screen.getText().concat("9")); c++;};
break;
			}
		}
		if(e.getActionCommand().equals("+"))
		{
			val = val + Integer.parseInt(screen.getText());
			System.out.println(Integer.parseInt(screen.getText()));
			screen.setText("" + val);
			c = 0;
			bool = true;
		}
		if(e.getActionCommand().equals("C")) {screen.setText(""); c = 0;
/*first_num = 0*/;}
	}

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



Question 8:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BaseConverter implements KeyListener{
	JFrame frame = new JFrame("Base Converter");
	String[] Base = new String[]{"Duodecimal", "Hexidecimal", "Decimal", "Octal",
"Binary"};
	int b1, b2;
	char key;
	JComboBox<String> drop1 = new JComboBox<>(Base), drop2 = new
JComboBox<>(Base);
	JTextField in = new JTextField(), out = new JTextField();

	public BaseConverter()
	{
		frame.setSize(400, 200);
		frame.setLayout(new GridLayout(2, 2, 20, 20));
		frame.add(in);
		in.addKeyListener(this);
		frame.add(out);
		out.addKeyListener(this);
		frame.add(drop1);
		frame.add(drop2);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
```

```
	public void keyPressed(KeyEvent e)
	{
		key = e.getKeyChar();
		valid();
	}

	public void keyTyped(KeyEvent e)
	{
		if(valid()==false)
		{
			e.consume();
			return;
		}
	}

	public void keyReleased(KeyEvent e)
	{
		convert();
	}

	public void displayInfo(KeyEvent e, String keyStatus){}

	public boolean valid(){
		switch(b1)
		{
		case 12: if(key >= '0' &&  key <= '9' || key == 'a' || key == 'b' || key
== 'A' || key == 'B')return true; break;
		case 16: if(key >= '0' && key <= '9' || key >= 'a' && key <='f' || key
>='A' && key <= 'F')return true; break;
		case 10: if(key >= '0' && key <= '9') return true; break;
		case 8: if(key >= '0' && key <= '7') return true; break;
		case 2: if(key >= '0' && key <= '1') return true; break;
		default: break;
		}
		return false;
	}
	public void convert()
	{
		switch(drop1.getSelectedItem().toString())
		{
		case "Duodecimal": b1 = 12; break;
		case "Hexidecimal": b1 = 16; break;
		case "Decimal": b1 = 10; break;
		case "Octal": b1 = 8; break;
		case "Binary": b1 = 2; break;
		default:break;
		}
		switch(drop2.getSelectedItem().toString())
		{
		case "Duodecimal": b2 = 12; break;
		case "Hexidecimal": b2 = 16; break;
		case "Decimal": b2 = 10; break;
		case "Octal": b2 = 8; break;
		case "Binary": b2 = 2; break;
```

```
                default:break;
                }
                out.setText(Long.toString(Long.parseLong(in.getText(), b1), b2));
        }

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