

# COMPUTATIONAL PHYSICS (SSP2122)

# Building a physics solver using Netbeans

## by Abd. Khamim Ismail Mohd Khalid Kasmin





### How to create a user interface (UI) using Netbeans

Based on the "GravityAttract07a.java" source code (shown below), build a user interface using Netbeans. The design of the UI could be look like this (**Figure1**);

A group of particles in two dimensional coordinate system are shown in the figure on the left. The mass and the coordinate of each particle are given. This program solves the net (resultant) gravitational force on any one of them due to others by using the principle of superposition. This is a general	5_ 4-	5- 4-			•	m <sub>5</sub> (4.2,4.5) 6.6 kg		
principle that says a net effect is the sum of the individual effects.	3 2 1	2.5 k	9 g m <sub>2</sub> (1.) 5 kg			9.2 I		
	+	1	2	3	4	5	→ X	
n which particle (m <sub>1</sub> m <sub>5</sub> ) you want to calculate ne resultant force on particle due to the ot		-	e?	(1,2,	5) N	J,		

Figure 1



### OPENCOURSEWARE

### "GravityAttract07a.java" source code

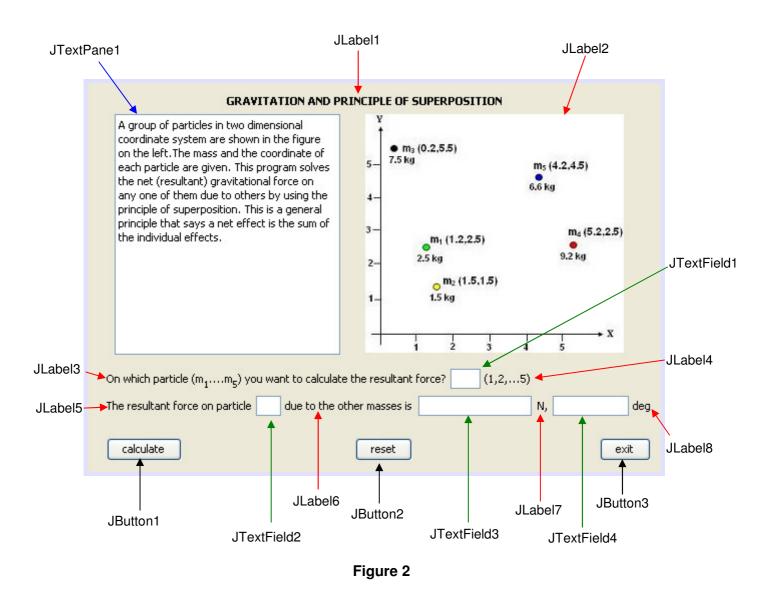
```
Java class for calculating the resultant force on any object
   (i.e either on m1,m2, m3, m4, or m5)
*/
import static java.lang.Math.*;
import java.io.*;
public class GravityAttract07a
  public static void main(String args[])
    double[] m = new double[6]; // mass of object
    double[] x = new double[6]; // x-coor of object
    double[] y = new double[6]; // y-coor of object
                         // resultant force
    double f;
    double fx,fy;
                          // x & y component of force
    double rx, ry, rp, co, G, dirf, angdirf;
    int j;
    G = 6.67e-11;
    m[1]=2.5; x[1]=1.2; y[1]=2.5;
    m[2]=1.5; x[2]=1.5; y[2]=1.5;
    m[3]=7.5; x[3]=0.2; y[3]=5.5;
    m[4]=9.2; x[4]=5.2; y[4]=2.5;
    m[5]=6.6; x[5]=4.2; y[5]=4.5;
    BufferedReader dataIn=new BufferedReader(new InputStreamReader(System.in));
    String mass="";
    System.out.println("On which mass you want to calculate the net force(1,2,3,4 or 5)?");
    try { mass=dataIn.readLine();}
    catch(IOException e){System.out.println("Not a valid input value!!");}
    int mi = Integer.parseInt(mass);
    fx = 0.0;
    fy = 0.0;
      for(j=1; j<=5; j++)
      {
        if (j!=mi)
        {
        rx = x[i]-x[mi];
        ry = y[j]-y[mi];
        rp = pow((rx^*rx+ry^*ry), 1.5); // (rx^2 + ry^2) raised to the power of 3/2
        co = G^{m[mi]m[i]};
        fx = fx + co^*rx/rp;
        fy = fy + co^* ry/rp;
      }
    f = sqrt(fx^*fx + fy^*fy);
    dirf = toDegrees(atan(fy/fx));
    System.out.printf("The force on m["+ mi +"] due to the other masses is %eN, %fdeg\n",f,dirf);
 }
```

}



This example will not show you step by step details procedure, since it already explained in the class and in the e-learning. Please refer to the e-learning about how to create a project, how to create JFrame Form and as well how to create JButton, JTextField, JLabel, JTextPane and so on.

First, create a new project and give a name e.g. "gravity" and then create JFrame Form and the following JButton, JTextField, JLabel, JTextPane etc. by using "swing palette" as shown in **Figure 2**.



To insert a picture into JLabel2; click on the JLabel2, go to properties and select "icon". On the icon editor choose file and then select file to locate the directory of your graphic/picture file (**Figure 3**). Select the file and click OK when finished. Make sure you have already created your own picture file and save it as .jpg, .png, .bmp etc. format e.g. "gravity.jpg". Use any graphic package to draw your diagram/picture or use drawing tools available in Open Office or Microsoft Word.



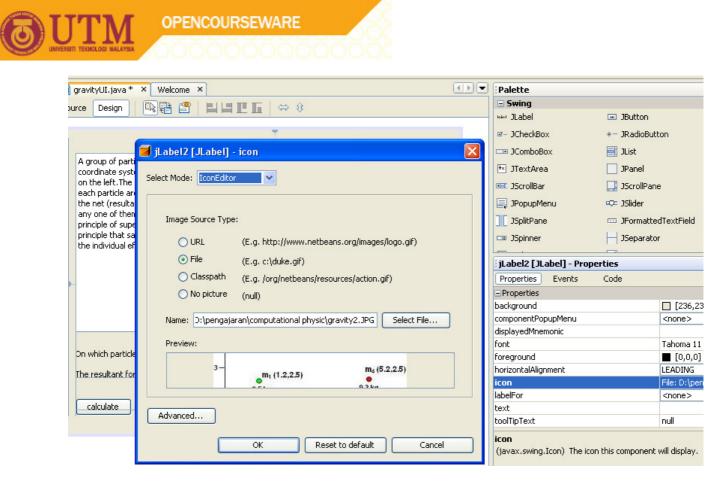


Figure 3

To write documentation or question, e.g. in JTextPane1; click on the JTextPane1, go to properties and select "text". Type the documentation in the text editor as shown in **Figure 4**. Click OK when finished.

		🖃 Swing		
extPane1 [JTextPane] - text 🛛 💦	<b>7</b> –	New JLabel	💌 JButton	甅 JTo
		Image: Image	IRadioButton	°∃ But
t Mode: StringEditor 💙		🗔 JComboBox	📑 JList	🗔 JTe
		* JTextArea	JPanel	<b>61</b> эта
		I JScrollBar	JScrollPane	File JM
group of particles in two dimensional coordinate system are shown in the figure on the left. The mass nd the coordinate of each particle are given. This program solves the net (resultant) gravitational				
orce on any one of them due to others by using the principle of superposition. This is a general		🔄 JPopupMenu	🔍 JSlider	📼 JPr
principle that says a net effect is the sum of the individual effects.		][] JSplitPane	📼 JFormattedTextField	📼 JP
		🚥 JSpinner	JSeparator	тл
		jTextPane1 [JTextPa	Code	
		Properties Events	Lode	
		background	[255,25	5.2551
		border	[MarginBorg	
		componentPopupMenu	<none></none>	-
		document	[DefaultSty	ledDocume
		editable		
		font foreground	Tahoma 11	Plain
		text	[0,0,0] A group of	narticles in
	toolTipText Other Prop text		null	paracies il
		Other Properties		
		<b>text</b> (java.lang.String) the tex	<t component<="" of="" td="" this=""><td></td></t>	
		Output		
Vanced				
OK Reset to default Cancel				







To do formatting on the text such as superscript and subscript e.g. in jLabel3; click on the JLabel3, go to properties and select "text". Type the documentation/message on the text editor as shown in **Figure 5** as follows;

### e.g.

### <html>On which particle (m<sub>1</sub>....m<sub>5</sub>) you want to calculate the resultant force?

Use **sup** instead, if you want a superscript. The result will look like this;

On which particle  $(m_1, \dots, m_p)$  you want to calculate the resultant force?

Click OK when finished.

<ul> <li>→ JRadioButton</li> <li>→ JRadioButton</li> <li>→ JRadioButton</li> <li>⇒ But</li> <li>⇒ JComboBox</li> <li>⇒ JLi</li> <li>⇒ JTextField</li> <li>⇒ JTextField&lt;</li></ul>	
Attml>On which particle (m <sub>1</sub> m <sub>5</sub> ) you want to calculate the resultant force?          Image: Control of the sub and the sub	
A thml>On which particle (m <sub>1</sub> m <sub>5</sub> ) you want to calculate the resultant force?          Image: Control of the sub and the s	
force?	
I I ToggleButton III - JC I I JToggleButton © - JC I I JComboBox III JL I JTextField III III I JPanel IIII IIII I JPanel IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	utton
JComboBox II JLi JTextField II JTextField II	heckBox
Description of the second seco	ttonGroup
☐ JPanel	st
I JScrollBar I JSc I III JMenuBar I JP	extArea
Eite JMenuBar 📃 JPr	abbedPane
	rollPane
iii shel2 [1] shel] _ Deeparties	opupMenu
Properties Events Code	
	6,233,216]
componentPopupMenu <none< th=""><th></th></none<>	
displayedMnemonic	
font Tahom	a 11 Plain
foreground	0,0]
horizontalAlignment LEADII	١G
icon null	
labelFor <none< td=""><td>&gt;</td></none<>	>
text <hr/>	∍On which part
toolTipText null	
Advanced         Advanced         Advanced	ext this compo
tput	
OK Reset to default Cancel	

Figure 5

To do the calculation, you need to put the coding similar like "GravityAttract07a.java" and put it into the "calculate" button (this is "JButton1"). You can copy "GravityAttract07a.java" source code,





OPENCOURSEWARE

paste it and then do a few alterations. First, click on the "calculate" button, right click, choose Events and then choose Action then actionPerformed (**Figure 6**). It will bring you to where you should insert your coding (**Figure 7**). You can write your own coding or copy/paste any coding from somewhere else (in this case from "GravityAttract07a.java").

		IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
GRAVITATION AN	D PRINCIPLE OF SUPERPOSITION	JComboBox
A group of particles in two dimensional coordinate system are shown in the figure	• m₃ (0.2,5.5)	🐄 JTextArea
on the left. The mass and the coordinate of each particle are given. This program solves	5_ 7.5 kg m-(4.2.4.5)	I IScrollBar
the net (resultant) gravitational force on	Action	1ActionPerformed]
any one of them due to others by using the principle of superposition. This is a general principle that says a net effect is the sum of the individual effects. On which particle (m <sub>1</sub> m <sub>5</sub> ) you want to calcu The resultant force on particle is Edit Text Change Variable Name	Ancestor Change Component Container Container Container Focus Hierarchy HierarchyBounds InputMethod Item Key Mouse Mouse MouseWheel PropertyChange Kov Ng m <sub>4</sub> (5.2,2.5) Sov Ng m <sub>4</sub> (5.2,2.5) Sov Ng (1,2,5) (1,2,5) exit	JSplitPane JSpinner JSpinner JButton1 [JButto Properties Eve Properties action background componentPopupMe font foreground icon mnemonic text toolTipText
Align Anchor Auto Resizing	VetoableChange	Other Properties jButton1 [JButto
Same Size	▶ ults	Output
Set Default Size		, output
Space Around Component		
Move Up Move Down		
Cut Ct	rl+X	
	rl+C	
	elete	
Properties		

Figure 6





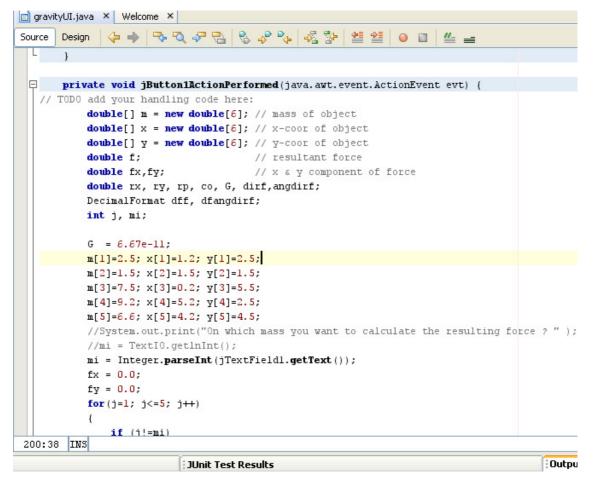


Figure 7

Copy "GravityAttract07a.java" source code and paste to the place after "// TODO add your handling code here:" see **Figure 7**. To see which part has been modified, the coding in **Figure 7** has been color coded for better understanding as follows;

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
     double[] m = new double[6]; // mass of object
     double[] x = new double[6]; // x-coor of object
     double[] y = new double[6]; // y-coor of object
     double f:
                          // resultant force
     double fx.fy:
                           // x & y component of force
     double rx, ry, rp, co, G, dirf, angdirf;
     DecimalFormat dff, dfangdirf;
     int j, mi;
     G = 6.67e-11;
     m[1]=2.5; x[1]=1.2; y[1]=2.5;
     m[2]=1.5; x[2]=1.5; y[2]=1.5;
     m[3]=7.5; x[3]=0.2; y[3]=5.5;
     m[4]=9.2; x[4]=5.2; y[4]=2.5;
     m[5]=6.6; x[5]=4.2; y[5]=4.5;
     // BufferedReader dataIn=new BufferedReader(new InputStreamReader(System.in));
     // String mass="";
     // System.out.println("On which mass you want to calculate the net force(1,2,3,4 or 5)?");
     // try { mass=dataIn.readLine();}
```



```
// catch(IOException e){System.out.println("Not a valid input value!!");}
             // int mi = Integer.parseInt(mass);
             mi = Integer.parseInt(jTextField1.getText());
             fx = 0.0;
             fy = 0.0;
             for(j=1; j<=5; j++)
             ł
                if (j!=mi)
                {
                 rx = x[j]-x[mi];
                 ry = y[i]-y[mi];
                 rp = Math.pow((rx*rx+ry*ry), 1.5); // (rx^2 + ry^2) raised to the power of 3/2
                 co = G^{m[mi]m[i]}
                 fx = fx + co^*rx/rp;
                 fy = fy + co^* ry/rp;
                }
             f = Math.sqrt(fx*fx + fy*fy);
             dirf = Math.atan(fy/fx);
             angdirf = (dirf/(22.0/7.0))*180.0;
             //System.out.printf("The resultant force on particle ["+ mi +"] due to the other masses is
%1.4e N, %2.2f deg\n",f,angdirf);
             dff = new java.text.DecimalFormat("#.####E00");
             dfangdirf = new DecimalFormat("#0.00");
             iTextField2.setText("" + mi);
             jTextField3.setText("" + dff.format(f));
             jTextField4.setText("" + dfangdirf.format(angdirf));
          }
```

### Note:

Brown = automatically generated

Blue = the original code from "GravityAttract07a.java"

- Green = coding from "GravityAttract07a.java" which were being disabled due to incompatibilities with Netbeans
- Red = a new inserted coding; a few of them is to rephrase/substitute the green one

Javascripts command line use:

BufferedReader dataIn=new BufferedReader(new InputStreamReader(System.in)); String mass=""; System.out.println("On which mass you want to calculate the net force(1,2,3,4 or 5)?"); try { mass=dataIn.readLine();} catch(IOException e){System.out.println("Not a valid input value!!");} int mi = Integer.parseInt(mass);

to read any input from the keyboard, typed by the user while Netbeans uses e.g. "Integer.parseInt(jTextField1.getText())", etc. Javascripts command line uses "System.out.printf" to format the printed output while Netbeans uses more complicated way to do the same job (see below);

### Javascript command line:

System.out.printf("The resultant force on particle ["+ mi +"] due to the other masses is %1.4e N, %2.2f degn",f,angdirf);

### Netbeans:

```
dff = new java.text.DecimalFormat("#0.###E00");
dfangdirf = new DecimalFormat("#0.00");
```



jTextField2.setText("" + mi); jTextField3.setText("" + dff.format(f)); jTextField4.setText("" + dfangdirf.format(angdirf));

"dff" and "dfangdirf" are new variables that should be declared and added to format the numbers into decimal format i.e. "#0.###E00" to get a scientific format e.g. "2.0019E-10" and "#0.00" to get a decimal number e.g. "-15.70".

To use DecimalFormat you have to put "import java.text.DecimalFormat;" in the top of the coding as shown in **Figure 8**.



### Figure 8

To reset the calculation, you need to put coding into the "reset" button (this is "JButton2"). First, click on the "reset" button, right click, choose Events and then choose Action then actionPerformed (similar like in **Figure 6**). It will bring you to where you should insert your coding (**Figure 9**). Write the following coding (**Figure 9**);



#### Figure 9

To exit the program, you need to put coding into the "exit" button (this is "JButton3"). First, click on the "exit" button, right click, choose Events and then choose Action then actionPerformed (similar like in **Figure 6**). It will bring you to where you should insert your coding (**Figure 10**). Write the following coding (**Figure 10**);



Figure 10

10

 $\odot \odot \odot$ 

ocw.utm.m



Once you finished, now you can run and see the result. If everything is fine then the output will be look like this (**Figure 11**);

GRAVITATION AND PRINCIPLE A group of particles in two dimensional coordinate system are shown in the figure on the left.The mass and the coordinate of each particle are given. This program solves the net (resultant) gravitational force on any one of them due to others by using the principle of superposition. This is a general principle that says a net effect is the	$\begin{array}{c} Y \\ 5-\\ 7.5 \text{ kg} \\ 4-\\ 3-\\ 2-\\ 2-\\ 2-\\ 2.5 \text{ kg} \end{array} \begin{array}{c} m_3 (0.2,5.5) \\ m_5 (4.2,4.5) \\ 6.6 \text{ kg} \\ 6.6 \text{ kg} \\ 9.2 \text{ kg} \end{array}$
Sum of the individual effects. On which particle (m <sub>1</sub> m <sub>5</sub> ) you want to calculate th	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$

Figure 11

Remember when you compile and run "GravityAttract07a.java" using Javascripts command line, once you compile and run, the result was looked like this;

On which particle you want to calculate the resulting force ? 2 The resultant force on particle [2] due to the other masses is 3.0673e-10 N, 88.37 deg

### SEE THE DIFFERENCES AT THE USER INTERFACE!!

