

Ramons II, Juanito M.  
Reyes, Trent Rufus N.  
CEIT-03-301P

Yotube Link: <https://youtu.be/fOWiBGo5V98>

MENU CODE SCREENSHOT:

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionListener;
4 import java.awt.event.ActionEvent;
5
6 class Menu2 extends JFrame implements ActionListener
7 {
8
9     JLabel l1 = new JLabel("SCHOOL SUPPLIES:");
10    JLabel l2 = new JLabel("BOOKS:");
11    JLabel l3 = new JLabel("NONFICTION BOOKS:");
12    JLabel l4 = new JLabel("ART MATERIALS:");
13
14    JCheckBox cb1 = new JCheckBox("PENS(ANY COLOR) - Php 10");
15    JCheckBox cb2 = new JCheckBox("PENCILS - Php 10");
16    JCheckBox cb3 = new JCheckBox("ERASERS - Php 19");
17    JCheckBox cb4 = new JCheckBox("FILING AND STORAGE - Php 20");
18    JCheckBox cb5 = new JCheckBox("PAPERS - Php 25/SET");
19    JCheckBox cb6 = new JCheckBox("FICTION BOOKS (ANY) - Php 500");
20    JCheckBox cb7 = new JCheckBox("NON-FICTION BOOKS (ANY) - Php 500");
21    JCheckBox cb15 = new JCheckBox("CANVAS - Php 200");
22    JCheckBox cb8 = new JCheckBox("SKETCH PAD - Php 180");
23    JCheckBox cb9 = new JCheckBox("PAINT/COLOR MATERIALS - Php 100/SET");
24
25
26    JLabel l15 = new JLabel("Product ID: SS001");
27    JLabel l16 = new JLabel("Product ID: SS002");
28    JLabel l17 = new JLabel("Product ID: SS003");
29    JLabel l18 = new JLabel("Product ID: SS004");
30    JLabel l19 = new JLabel("Product ID: SS005");
31    JLabel l110 = new JLabel("Product ID: BK001");
32    JLabel l111 = new JLabel("Product ID: BK002");
33    JLabel l112 = new JLabel("Product ID: AM001");
34    JLabel l113 = new JLabel("Product ID: AM002");
35    JLabel l114 = new JLabel("Product ID: AM003/AM004");
36
37
38    JButton b = new JButton("Order");
39    JButton p1 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\pens.png"));
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
37
38 JButton b = new JButton("Order");
39 JButton p1 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\pens.png"));
40 JButton p2 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\pencils.png"));
41 JButton p3 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\erasers.png"));
42 JButton p4 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\filing and storage.png"));
43 JButton p5 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\PAPER.png"));
44 JButton p6 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\fiction (2).png"));
45 JButton p7 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\nonfiction.png"));
46 JButton p8 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\canvas.png"));
47 JButton p9 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\sketchpad.png"));
48 JButton p10 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\paintmaterial.png"));
49 String color[]={ "red", "blue", "green", "black" };
50 String type[]={ "graphite", "charcoal" };
51 String type1[]={ "vinyl", "rubber", "gum eraser", "sand eraser" };
52 String type2[]={ "long folder", "short folder", "long envelope", "short envelope", "mini bookshelf", "plastic envelope" };
53 String type3[]={ "short bond paper", "colored paper", "oslo paper", "long bond paper", "yellow pad", "intermediate pad", "notebook" };
54 String type4[]={ "Fable", "Fairy Tale",
55     "Fantasy",
56     "Folklore",
57     "Horror",
58     "Humor",
59     "Legend",
60     "Mystery",
61     "Mythology",
62     "Science Fiction",
63     "Short Story",
64     "Tall Tale" };
65 String type5[]={ "Biography",
66     "Elementary Education",
67     "JHS Education",
68     "SHS Education",
69     "Tertiary Education",
70     "Autobiography",
71     "Novel",
72     "World History",
73     "Phillipine History" };
74 String type6[]={ "Universal-10x8", "Universal-20x16", "Oil-10x8", "Oil-20x16" };
75 String type7[]={ "7x10", "9x12", "11x14", "18x24", "24x36" };
```

```

75 String type7[]= {"7x10","9x12","11x14","18x24","24x36"};
76 String type8[]= {"Paint-Red","Paint-Yellow","Paint-Blue","Paint-Black","Paint-White","Paint-Assorted", "Color Pencil-Assorted",
77 "Color Pencil-Red","Color Pencil-Yellow","Color Pencil-Blue","Color Pencil-Black","Color Pencil-White"};
78
79 JTextField tf1=new JTextField();
80 JTextField tf2=new JTextField();
81 JTextField tf3=new JTextField();
82 JTextField tf4=new JTextField();
83 JTextField tf5=new JTextField();
84 JTextField tf6=new JTextField();
85 JTextField tf7=new JTextField();
86 JTextField tf8=new JTextField();
87 JTextField tf9=new JTextField();
88 JTextField tf10=new JTextField();
89 JLabel t1 = new JLabel("Color: ");
90 JLabel t2 = new JLabel("Type: ");
91 JLabel t3 = new JLabel("Type: ");
92 JLabel t4 = new JLabel("Type: ");
93 JLabel t5 = new JLabel("Type: ");
94 JLabel t6 = new JLabel("Type: ");
95 JLabel t7 = new JLabel("Type: ");
96 JLabel t8 = new JLabel("Type/Size: ");
97 JLabel t9 = new JLabel("Size: ");
98 JLabel t10 = new JLabel("Type/Color: ");
99 JLabel q1 = new JLabel("Quantity: ");
100 JLabel q2 = new JLabel("Quantity: ");
101 JLabel q3 = new JLabel("Quantity: ");
102 JLabel q4 = new JLabel("Quantity: ");
103 JLabel q5 = new JLabel("Quantity: ");
104 JLabel q6 = new JLabel("Quantity: ");
105 JLabel q7 = new JLabel("Quantity: ");
106 JLabel q8 = new JLabel("Quantity: ");
107 JLabel q9 = new JLabel("Quantity: ");
108 JLabel q10 = new JLabel("Quantity: ");
109
110 final JComboBox combobox1 = new JComboBox(color);
111 final JComboBox combobox2 = new JComboBox(type);
112 final JComboBox combobox3 = new JComboBox(type1);
113 final JComboBox combobox4 = new JComboBox(type2);

```

```

109
110 final JComboBox combobox1 = new JComboBox(color);
111 final JComboBox combobox2 = new JComboBox(type);
112 final JComboBox combobox3 = new JComboBox(type1);
113 final JComboBox combobox4 = new JComboBox(type2);
114 final JComboBox combobox5 = new JComboBox(type3);
115 final JComboBox combobox6 = new JComboBox(type4);
116 final JComboBox combobox7 = new JComboBox(type5);
117 final JComboBox combobox8 = new JComboBox(type6);
118 final JComboBox combobox9 = new JComboBox(type7);
119 final JComboBox combobox10 = new JComboBox(type8);
120
121
122 public Menu2()
123 {
124     setLayout();
125     setLocationAndSize();
126     setColorAndFonts();
127     components();
128     addActionEvent();
129 }
130
131 public void setLayout()
132 {
133     setLayout(null);
134     setVisible(true);
135 }
136
137 public void setLocationAndSize()
138 {
139     l1.setBounds(550,0,700,60);
140
141     cb1.setBounds(30,45,190,20);
142     p1.setBounds(55, 70, 150, 150);
143     combobox1.setBounds(90, 230, 90, 20);
144     t1.setBounds(30, 230, 90, 20);
145     tf1.setBounds(90,260,150,20);
146     q1.setBounds(30, 260, 150, 20);
147     l5.setBounds(30, 290, 150, 20);

```

```
147     15.setBounds(30, 290, 150, 20);
148
149     cb2.setBounds(290,45,150,20);
150     p2.setBounds(315,70,150,150);
151     combobox2.setBounds(350,230, 90, 20);
152     t2.setBounds(290, 230, 90, 20);
153     tf2.setBounds(350,260,150,20);
154     q2.setBounds(290, 260, 150, 20);
155     l6.setBounds(290, 290, 150, 20);
156
157     cb3.setBounds(550,45,150,20);
158     p3.setBounds(575,70,150,150);
159     combobox3.setBounds(610,230, 90, 20);
160     t3.setBounds(550, 230, 90, 20);
161     tf3.setBounds(610,260,150,20);
162     q3.setBounds(550, 260, 150, 20);
163     l7.setBounds(550, 290, 150, 20);
164
165     cb4.setBounds(810,45,210,20);
166     p4.setBounds(835,70,150,150);
167     combobox4.setBounds(870,230, 120, 20);
168     t4.setBounds(810, 230, 90, 20);
169     tf4.setBounds(870,260,150,20);
170     q4.setBounds(810, 260, 150, 20);
171     l8.setBounds(810, 290, 150, 20);
172
173     cb5.setBounds(1070,45,150,20);
174     p5.setBounds(1095,70,150,150);
175     combobox5.setBounds(1130,230, 140, 20);
176     t5.setBounds(1070, 230, 90, 20);
177     tf5.setBounds(1130,260,150,20);
178     q5.setBounds(1070, 260, 150, 20);
179     l9.setBounds(1070, 290, 150, 20);
180
181     l2.setBounds(10,310,700,60);
182
183     cb6.setBounds(30,355,250,20);
184     p6.setBounds(55,380,150,150);
185     combobox6.setBounds(90,540, 120, 20);
```

```
183     cb6.setBounds(30,355,250,20);
184     p6.setBounds(55,380,150,150);
185     combobox6.setBounds(90,540, 120, 20);
186     t6.setBounds(30, 540, 90, 20);
187     tf6.setBounds(90,570,150,20);
188     q6.setBounds(30, 570, 150, 20);
189     l10.setBounds(30, 600, 150, 20);
190
191     cb7.setBounds(290,355,250,20);
192     p7.setBounds(315,380,150,150);
193     combobox7.setBounds(350,540, 120, 20);
194     t7.setBounds(290, 540, 90, 20);
195     tf7.setBounds(350,570,150,20);
196     q7.setBounds(290, 570, 150, 20);
197     l11.setBounds(290, 600, 150, 20);
198
199     l4.setBounds(550,310,700,60);
200     p8.setBounds(575,380,150,150);
201     cb15.setBounds(550,355,180,20);
202     combobox8.setBounds(610,540, 120, 20);
203     t8.setBounds(530, 540, 90, 20);
204     tf8.setBounds(610, 570,150,20);
205     q8.setBounds(530, 570, 150, 20);
206     l12.setBounds(530, 600, 150, 20);
207
208     cb8.setBounds(810,355,180,20);
209     p9.setBounds(835,380,150,150);
210     combobox9.setBounds(870,540, 120, 20);
211     t9.setBounds(810, 540, 90, 20);
212     tf9.setBounds(870, 570,150,20);
213     q9.setBounds(810, 570, 150, 20);
214     l13.setBounds(810, 600, 150, 20);
215
216     cb9.setBounds(1050,355,285,20);
217     p10.setBounds(1095,380,150,150);
218     combobox10.setBounds(1140,540, 150, 20);
219     t10.setBounds(1070, 540, 90, 20);
220     tf10.setBounds(1140, 570,150,20);
221     q10.setBounds(1070, 570, 150, 20);
```



```

219         t10.setBounds(1070, 540, 90, 20);
220         tf10.setBounds(1140, 570,150,20);
221         q10.setBounds(1070, 570, 150, 20);
222         l14.setBounds(1070, 600, 150, 20);
223
224         b.setBounds(600,680,80,30);
225     }
226
227     public void setColorAndFonts()
228     {
229         Color c1 = new Color(241, 161, 97);
230         l1.setFont(new Font("Arial", Font.BOLD, 25));
231         l1.setForeground(c1);
232         l2.setFont(new Font("Arial", Font.BOLD, 25));
233         l2.setForeground(c1);
234         l4.setFont(new Font("Arial", Font.BOLD, 25));
235         l4.setForeground(c1);
236
237         b.setBackground(c1);
238         b.setForeground(Color.white);
239         b.setFont(new Font("Arial", Font.BOLD, 15));
240     }
241
242     public void components()
243     {
244         add(l1);
245         add(l2);
246         add(l4);
247         add(cb1);
248         add(cb2);
249         add(cb3);
250         add(cb4);
251         add(cb5);
252
253         add(cb6);
254         add(cb7);
255         add(cb15);
256         add(cb8);
257         add(cb9);

```

```

257         add(cb9);
258
259         add(b);
260         add(p1);
261         add(p2);
262         add(p3);
263         add(p4);
264         add(p5);
265         add(p6);
266         add(p7);
267         add(p8);
268         add(p9);
269         add(p10);
270         add(combobox1);
271         add(combobox2);
272         add(combobox3);
273         add(combobox4);
274         add(combobox5);
275         add(combobox6);
276         add(combobox7);
277         add(combobox8);
278         add(combobox9);
279         add(combobox10);
280         add(tf1);
281         add(tf2);
282         add(tf3);
283         add(tf4);
284         add(tf5);
285         add(tf6);
286         add(tf7);
287         add(tf8);
288         add(tf9);
289         add(tf10);
290         add(q1);
291         add(q2);
292         add(q3);
293         add(q4);
294         add(q5);
295         add(q6);

```

```

295     add(q6);
296     add(q7);
297     add(q8);
298     add(q9);
299     add(q10);
300
301     add(t1);
302     add(t2);
303     add(t3);
304     add(t4);
305     add(t5);
306     add(t6);
307     add(t7);
308     add(t8);
309     add(t9);
310     add(t10);
311
312     add(l5);
313     add(l6);
314     add(l7);
315     add(l8);
316     add(l9);
317     add(l10);
318     add(l11);
319     add(l12);
320     add(l13);
321     add(l14);
322 }
323
324 public void addActionEvent()
325 {
326     b.addActionListener(this);
327 }
328
329 @Override
330 public void actionPerformed(ActionEvent e)
331 {
332     float amount = 0;
333     String msg = "";

```

```

333     String msg = "";
334
335     if(cb1.isSelected())
336     {
337         String s1=tf1.getText();
338         int a =Integer.parseInt(s1);
339         amount+=10*a;
340         msg+="PEN: " + a + "\n" + "Color: " + combobox1.getItemAt(combobox1.getSelectedIndex()) + "\n";
341     }
342
343     if(cb2.isSelected())
344     {
345         String s2=tf2.getText();
346         int b=Integer.parseInt(s2);
347         amount+=10*b;
348         msg+="PENCILS: " + b + "\n" + "Type: " + combobox2.getItemAt(combobox2.getSelectedIndex())+ "\n";
349     }
350
351     if(cb3.isSelected())
352     {
353         String s3=tf3.getText();
354         int c=Integer.parseInt(s3);
355         amount+=19*c;
356         msg+="ERASER: " + c + "\n" + "Type: " + combobox3.getItemAt(combobox3.getSelectedIndex())+ "\n";
357     }
358
359     if(cb4.isSelected())
360     {
361         String s4=tf4.getText();
362         int d=Integer.parseInt(s4);
363         amount+=20*d;
364         msg+="FILING AND STORAGE: " + d + "\n" + "Type: " + combobox4.getItemAt(combobox4.getSelectedIndex())+ "\n";
365     }
366
367     if(cb5.isSelected())
368     {
369         String s5=tf5.getText();
370         int f=Integer.parseInt(s5);
371         amount+=25*f;
372         msg+="PAPERS: " + f + "\n" + "Type: " + combobox5.getItemAt(combobox5.getSelectedIndex())+ "\n";
373     }

```

```

371     }
372
373     if(cb6.isSelected())
374     {
375         String s6=tf6.getText();
376         int g=Integer.parseInt(s6);
377         amount+=500*g;
378         msg+="FICTION BOOK: " + g + "\n" + "Type: " + combobox6.getItemAt(combobox6.getSelectedIndex()) + "\n";
379     }
380
381
382     if(cb7.isSelected())
383     {
384         String s7=tf7.getText();
385         int h=Integer.parseInt(s7);
386         amount+=500*h;
387         msg+="NON-FICTION BOOK: " + h + "\n" + "Type: " + combobox7.getItemAt(combobox7.getSelectedIndex()) + "\n";
388     }
389
390     if(cb15.isSelected())
391     {
392         String s8=tf8.getText();
393         int i=Integer.parseInt(s8);
394         amount+=200*i;
395         msg+="CANVAS: " + i + "\n" + "Type: " + combobox8.getItemAt(combobox8.getSelectedIndex()) + "\n";
396     }
397     if(cb8.isSelected())
398     {
399         String s9=tf9.getText();
400         int j=Integer.parseInt(s9);
401         amount+=180*j;
402         msg+="SKETCH PAD: " + j + "\n" + "Type: " + combobox9.getItemAt(combobox9.getSelectedIndex()) + "\n";
403     }
404     if(cb9.isSelected())
405     {
406         String s10=tf10.getText();
407         int k=Integer.parseInt(s10);
408         amount+=100*k;
409         msg+="PAINT/COLOR MATERIALS: " + k + "\n" + "Type: " + combobox10.getItemAt(combobox10.getSelectedIndex()) + "\n";

```

```

395         msg+="CANVAS: " + i + "\n" + "Type: " + combobox8.getItemAt(combobox8.getSelectedIndex()) + "\n";
396     }
397     if(cb8.isSelected())
398     {
399         String s9=tf9.getText();
400         int j=Integer.parseInt(s9);
401         amount+=180*j;
402         msg+="SKETCH PAD: " + j + "\n" + "Type: " + combobox9.getItemAt(combobox9.getSelectedIndex()) + "\n";
403     }
404     if(cb9.isSelected())
405     {
406         String s10=tf10.getText();
407         int k=Integer.parseInt(s10);
408         amount+=100*k;
409         msg+="PAINT/COLOR MATERIALS: " + k + "\n" + "Type: " + combobox10.getItemAt(combobox10.getSelectedIndex()) + "\n";
410     }
411
412     msg+= "-----\n";
413     JOptionPane.showMessageDialog(this, msg+"Total: " + amount);
414
415 }
416
417
418 public class BookStore {
419     public static void main(String[] args) {
420
421         Menu2 menu2 = new Menu2();
422
423         menu2.setTitle("Products");
424         Color c3 = new Color(253, 238, 220);
425         menu2.getContentPane().setBackground(c3);
426         menu2.setSize(1350, 800);
427         menu2.setVisible(true);
428         menu2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
429         menu2.setLocationRelativeTo(null);
430         menu2.setResizable(false);
431     }
432 }
433

```


OUTPUT:

MENU:

Products

SCHOOL SUPPLIES:

PENS(ANY COLOR) - Php 10




Color: red

Quantity:

Product ID: SS001

PENCILS - Php 10




Type: graphite

Quantity:

Product ID: SS002

ERASERS - Php 19




Type: vinyl

Quantity:

Product ID: SS003

FILING AND STORAGE - Php 20




Type: long folder

Quantity:

Product ID: SS004

PAPERS - Php 25/SET




Type: short bond paper

Quantity:

Product ID: SS005

BOOKS:

FICTION BOOKS (ANY) - Php 500




Type: Fable

Quantity:

Product ID: BK001

NON-FICTION BOOKS (ANY) - Php 500




Type: Biography

Quantity:

Product ID: BK002

CANVAS- Php 200




Type/Size: Universal-10x8

Quantity:

Product ID: AM001

SKETCH PAD - Php 180




Size: 7x10

Quantity:

Product ID: AM002

PAINT/COLOR MATERIALS - Php 100/SET



Type/Color: Paint-Red

Quantity:

Product ID: AM003/AM004

Order

```

classDiagram
    class BookstoreProducts {
        - Stock: int
        - productName: String
        - productImage
        - productID: String
        - ageLimit: int
        - price: float
        - type: String
        - discount: int
        - shopNow()
        - getProductInfo()
        - getPrice()
        - ship()
        - deliver()
    }
    class SchoolSupplies {
        - Classification: String
        - Color
        - getClassification()
        - getColor()
    }
    class Books {
        - Publisher: String
        - Genre: String
        - Author: String
        - Title: String
        - DatePublished: String
        - DisplayInfo()
    }
    class ArtMaterials {
        - Color: String
        - Model: String
        - Material: String
        - Type: String
        - getColor()
        - getMaterial()
    }
    BookstoreProducts <|-- SchoolSupplies
    BookstoreProducts <|-- Books
    BookstoreProducts <|-- ArtMaterials
  
```

The diagram illustrates a class hierarchy for a bookstore. The base class, **Bookstore Products**, defines common attributes and methods for all products. It includes attributes like `Stock`, `productName`, `productImage`, `productID`, `ageLimit`, `price`, `type`, and `discount`. Methods include `shopNow()`, `getProductInfo()`, `getPrice()`, `ship()`, and `deliver()`. Three subclasses inherit from **Bookstore Products**: **School Supplies**, **Books**, and **Art Materials**. **School Supplies** has attributes `Classification` and `Color`, and methods `getClassification()` and `getColor()`. **Books** has attributes `Publisher`, `Genre`, `Author`, `Title`, and `DatePublished`, and a method `DisplayInfo()`. **Art Materials** has attributes `Color`, `Model`, `Material`, and `Type`, and methods `getColor()` and `getMaterial()`.

UML:

SCHOOL SUPPLIES:

PENS(ANY COLOR) - Php 10



Color: 

red

Quantity:

Product ID: SS001

PENCILS - Php 10



Type: 

graphite

Quantity:

Product ID: SS002

ERASERS - Php 19



Type: 

vinyl

Quantity:

Product ID: SS003

FILING AND STORAGE - Php 20



Type: 

long folder

Quantity:

Product ID: SS004

PAPERS - Php 25/SET



Type: 

short bond paper

Quantity:

Product ID: SS005

SCHOOL SUPPLIES:

UML:

```
classDiagram
    class SchoolSupplies {
        - Classification: String
        - Color
        - getClassification()
        - getColor()
    }
    class Pencil {
        - pencilID: String
        - pencilCore: String
        - getPencilInfo()
    }
    class Pen {
        - penID: String
        - InkType: String
        - getPenInfo()
    }
    class Eraser {
        - eraserID: String
        - shape: String
        - color: String
        - size: String
        - getEraserInfo()
    }
    class Paper {
        - paperID: String
        - paperType: String
        - paperSize: String
        - color: String
        - getPaperInfo()
    }
    class FilingAndStorage {
        - Size: String
        - storageID: String
        - storageType: String
        - getStorageInfo()
    }
    SchoolSupplies <|-- Pencil
    SchoolSupplies <|-- Pen
    SchoolSupplies <|-- Eraser
    SchoolSupplies <|-- Paper
    SchoolSupplies <|-- FilingAndStorage
```



The diagram illustrates a class hierarchy for school supplies. At the top is the **School Supplies** class, which serves as the base. It has two attributes: **- Classification: String** and **- Color**, and two methods: **- getClassification()** and **- getColor()**. Below it are five subclasses: **Pencil**, **Pen**, **Eraser**, **Paper**, and **Filing and Storage**. Each subclass inherits from the base class and has its own set of attributes and methods. **Pencil** has **- pencilID: String** and **- pencilCore: String** with **- getPencilInfo()**. **Pen** has **- penID: String** and **- InkType: String** with **- getPenInfo()**. **Eraser** has **- eraserID: String**, **- shape: String**, **- color: String**, and **- size: String** with **- getEraserInfo()**. **Paper** has **- paperID: String**, **- paperType: String**, **- paperSize: String**, and **- color: String** with **- getPaperInfo()**. **Filing and Storage** has **- Size: String**, **- storageID: String**, and **- storageType: String** with **- getStorageInfo()**.



## BOOKS:

### BOOKS:

☐ FICTION BOOKS (ANY) - Php 500
 ☐ NON-FICTION BOOKS (ANY) - Php 500

Type:

Quantity:

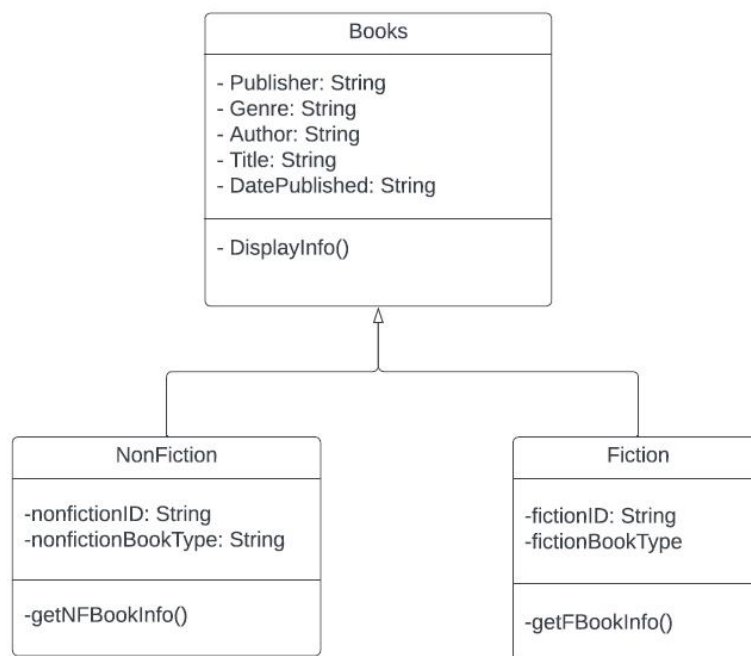
Product ID: BK001

Type:

Quantity:


Product ID: BK002

## UML:



## ART MATERIALS:

### ART MATERIALS:

☐ CANVAS- Php 200
 

Type/Size:

Quantity:


Product ID: AM001

☐ SKETCH PAD - Php 180
 

Size:

Quantity:

Product ID: AM002

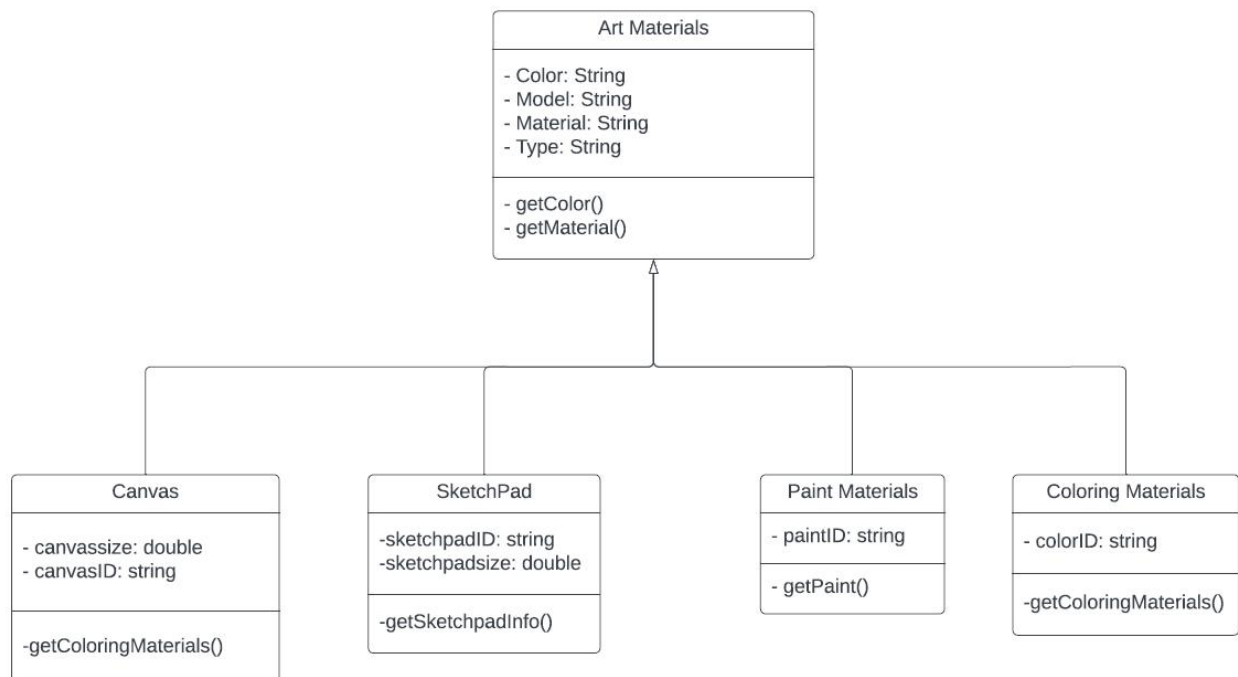
☐ PAINT/COLOR MATERIALS - Php 100/SET
 

Type/Color:

Quantity:

Product ID: AM003/AM004

## UML:



## LOGIN CODE:

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 class Login extends JFrame implements ActionListener {
7     Container container = getContentPane();
8     JLabel welcomeLabel = new JLabel("Welcome To Marahuyo Bookstore");
9     JLabel usernameLabel = new JLabel("Username:");
10    JLabel passwordLabel = new JLabel("Password:");
11    final JTextField user = new JTextField();
12    final JPasswordField pass = new JPasswordField();
13    JButton loginButton = new JButton("LOGIN");
14    JButton resetButton = new JButton("RESET");
15    JButton j1 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\FINAL LOGO.png"));
16    JButton j2 = new JButton(new ImageIcon("C:\\Users\\Juanito\\Desktop\\Acads\\Advanced Programming\\MENU PROGRAM\\final slogan.png"));
17    JCheckBox showPassword = new JCheckBox("Show Password");
18
19    Login()
20    {
21        setLayout();
22        setLocationAndSize();
23        setFontsAndColors();
24        addComponentsToContainer();
25        addActionEvent();
26    }
27
28    public void setLayout()
29    {
30        container.setLayout(null);
31        container.setVisible(true);
32    }
33
34    public void setLocationAndSize()
35    {
36        welcomeLabel.setBounds(30, 200, 700, 50);
37        usernameLabel.setBounds(100, 275, 140, 50);
38        user.setBounds(260, 275, 200, 40);
39        passwordLabel.setBounds(100, 335, 140, 30);
40
41        pass.setBounds(260, 335, 200, 40);
42        showPassword.setBounds(250, 380, 150, 30);
43        loginButton.setBounds(190, 440, 140, 60);
44        resetButton.setBounds(340, 440, 140, 60);
45        j1.setBounds(250, 30, 200, 150);
46        j2.setBounds(700, 0, 680, 570);
47    }
48
49    public void addComponentsToContainer()
50    {
51        container.add(welcomeLabel);
52        container.add(usernameLabel);
53        container.add(user);
54        container.add(passwordLabel);
55        container.add(pass);
56        container.add(showPassword);
57        container.add(loginButton);
58        container.add(resetButton);
59        container.add(j1);
60        container.add(j2);
61    }
62
63    public void setFontsAndColors()
64    {
65        welcomeLabel.setFont(new Font("Arial", Font.BOLD, 40));
66        Color c1 = new Color(241, 161, 97);
67        welcomeLabel.setForeground(c1);
68
69        usernameLabel.setBounds(100, 275, 140, 50);
70        usernameLabel.setFont(new Font("Arial", Font.BOLD, 25));
71
72        user.setBounds(260, 275, 200, 40);
73        user.setFont(new Font("Arial", Font.BOLD, 25));
74
75        passwordLabel.setBounds(100, 335, 140, 30);
```

```

73     user.setFont(new Font("Arial", Font.BOLD, 25));
74
75     passwordLabel.setBounds(100, 335, 140, 30);
76     passwordLabel.setFont(new Font("Arial", Font.BOLD, 25));
77
78     pass.setFont(new Font("Arial", Font.BOLD, 25));
79
80     showPassword.setBounds(255, 385, 160, 30);
81     showPassword.setFont(new Font("Arial", Font.BOLD, 15));
82     Color c2 = new Color(253,238,220);
83     showPassword.setBackground(c2);
84
85     loginButton.setFont(new Font("Arial", Font.BOLD, 25));
86     loginButton.setBackground(c1);
87     loginButton.setForeground(Color.white);
88
89     resetButton.setFont(new Font("Arial", Font.BOLD, 25));
90     resetButton.setBackground(c1);
91     resetButton.setForeground(Color.white);
92 }
93
94 public void addActionEvent()
95 {
96     loginButton.addActionListener(this);
97     resetButton.addActionListener(this);
98     showPassword.addActionListener(this);
99 }
100
101 int attempts = 1;
102
103 @Override
104 public void actionPerformed(ActionEvent e) {
105
106     boolean successLogin = true;
107
108     String userName = "Juanito";
109     String passWord = "abc123";
110
111     //Coding Part of LOGIN button

```

```

110
111     //Coding Part of LOGIN button
112
113     if (e.getSource() == loginButton)
114     {
115         do
116         {
117
118             String username = user.getText();
119             String password = new String(pass.getPassword());
120
121             JFrame f;
122             f = new JFrame();
123
124             if (userName.equals(username) && passWord.equals(password))
125             {
126                 JOptionPane.showMessageDialog(f, "LOGIN SUCCESSFUL!");
127                 new Menu1();
128             }
129             else if (userName.equals(username) && !(password.equals(password))) && attempts == 3)
130             {
131                 attempts++;
132                 JOptionPane.showMessageDialog(f, "You have reached that maximum number of attempts!");
133                 System.exit(0);
134             }
135             else if (!(userName.equals(username)) && !(password.equals(password))) && attempts == 3)
136             {
137                 attempts++;
138                 JOptionPane.showMessageDialog(f, "You have reached the maximum number of attempts!");
139                 System.exit(0);
140             }
141             else
142             {
143                 JOptionPane.showMessageDialog(f, "LOGIN FAILED! Attempt: " + attempts + "/3", "Warning", JOptionPane.WARNING_MESSAGE);
144                 attempts++;
145             }
146         }
147         while (!successLogin);
148

```



```

146     }
147     }
148     while (!successLogin);
149 }
150
151 //Coding Part of RESET button
152 if (e.getSource() == resetButton)
153 {
154     user.setText("");
155     pass.setText("");
156 }
157
158 //Coding Part of showPassword JCheckBox
159 if (e.getSource() == showPassword)
160 {
161     if (showPassword.isSelected())
162     {
163         pass.setEchoChar((char) 0);
164     }
165     else
166     {
167         pass.setEchoChar('*');
168     }
169 }
170
171 }
172
173 }
174 }
175
176 public class MarahuyoBookstore {
177     public static void main(String[] args)
178     {
179         Login frame = new Login();
180         frame.setTitle("MARAHUYO BOOKSTORE");
181         Color c3 = new Color(253,238,220);
182         frame.getContentPane().setBackground(c3);
183         frame.setSize(1390,570);
184         frame.setVisible(true);

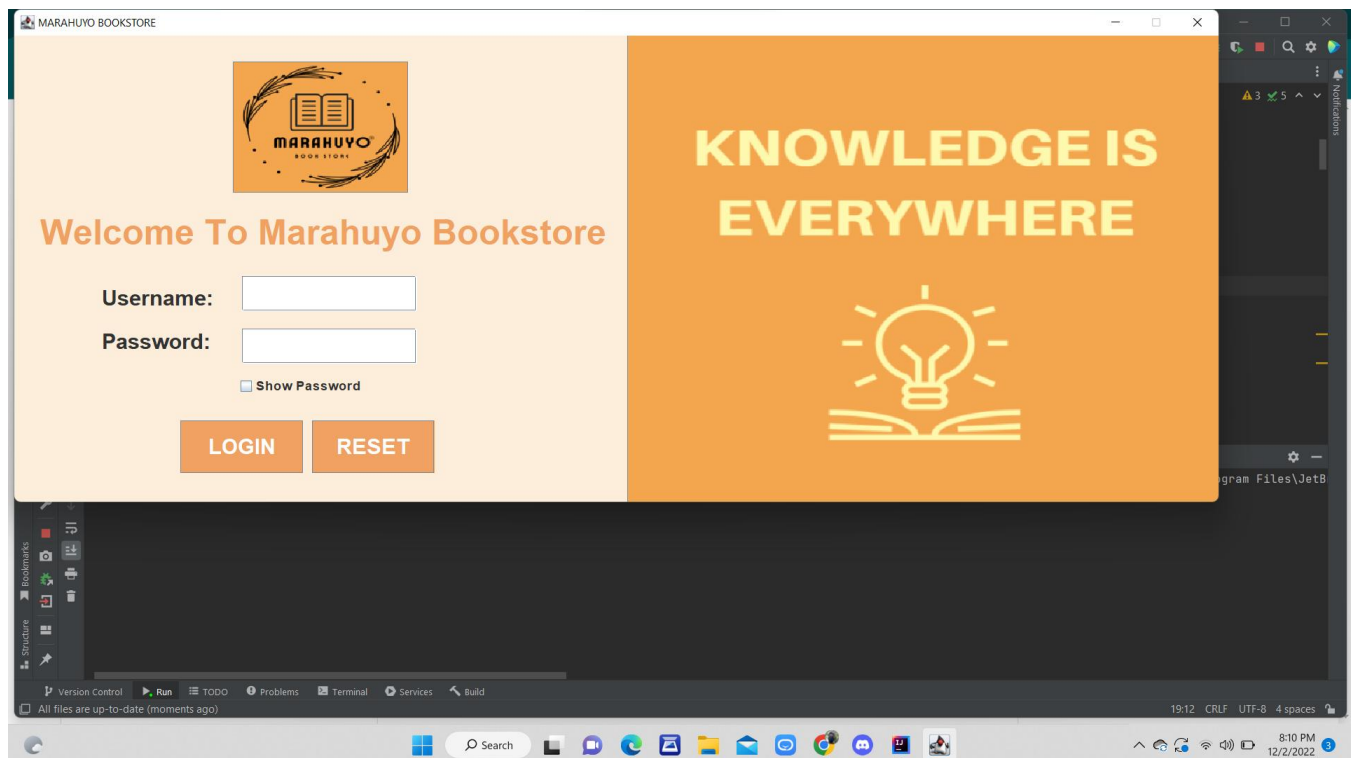
```

```

173     }
174 }
175
176 public class MarahuyoBookstore {
177     public static void main(String[] args)
178     {
179         Login frame = new Login();
180         frame.setTitle("MARAHUYO BOOKSTORE");
181         Color c3 = new Color(253,238,220);
182         frame.getContentPane().setBackground(c3);
183         frame.setSize(1390,570);
184         frame.setVisible(true);
185         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
186         frame.setResizable(false);
187     }
188 }

```

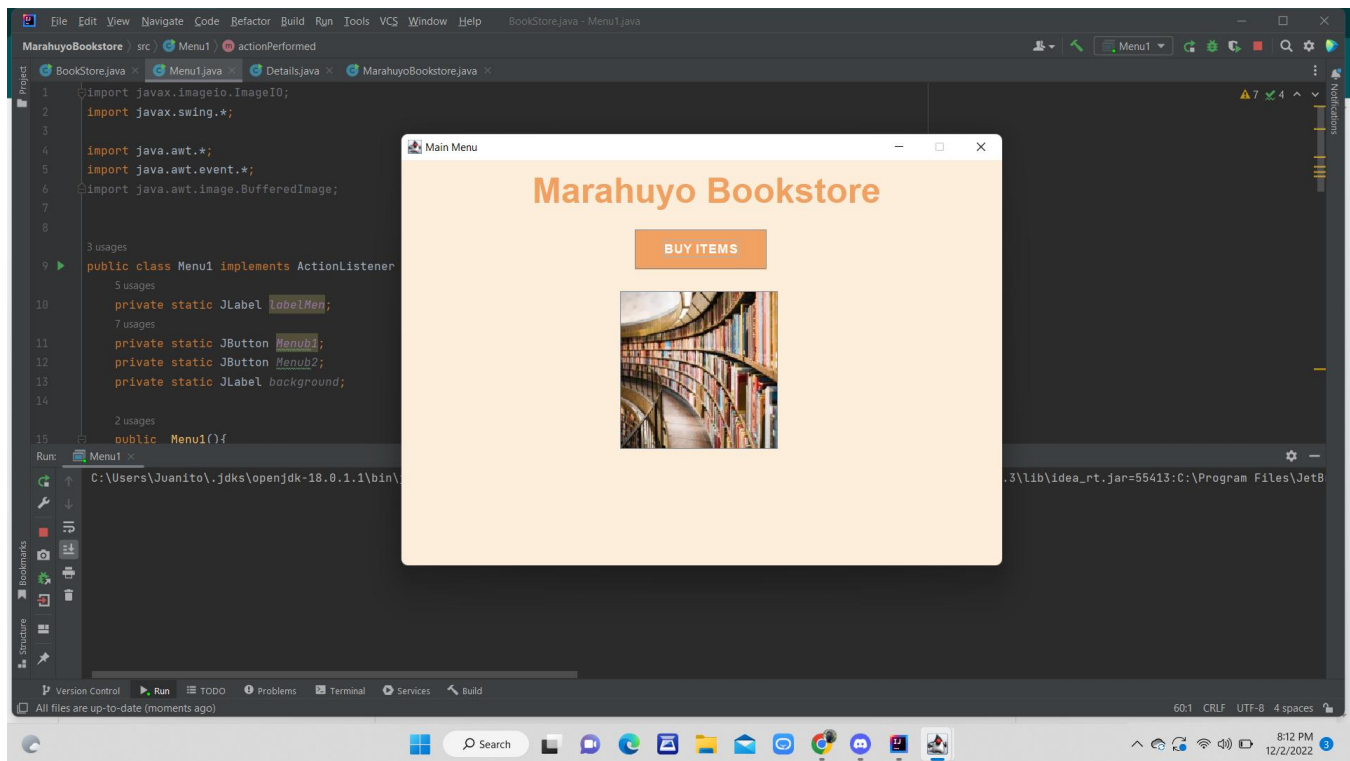
OUTPUT:



## MENU CODE:

```
1 import javax.imageio.ImageIO;
2 import javax.swing.*;
3
4 import java.awt.*;
5 import java.awt.event.*;
6 import java.awt.image.BufferedImage;
7
8
9 public class Menu1 implements ActionListener {
10     private static JLabel LabelMen;
11     private static JButton Menub1;
12     private static JButton Menub2;
13     private static JLabel background;
14
15     public Menu1(){
16         JPanel panelMen = new JPanel();
17         JFrame frameMen = new JFrame();
18
19         frameMen.setSize(700,500);
20         frameMen.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         frameMen.add(panelMen);
22         frameMen.setTitle("Main Menu");
23         frameMen.setLocationRelativeTo(null);
24         frameMen.setResizable(false);
25         frameMen.setVisible(true);
26
27         panelMen.setLayout(null);
28         panelMen.setBackground(new Color(253,238,220));
29
30         LabelMen = new JLabel("Marahuyo Bookstore");
31         LabelMen.setBounds(200,10,700,50);
32         LabelMen.setFont(new Font("Arial", Font.BOLD, 30));
33         LabelMen.setForeground(new Color(241, 161, 97));
34         panelMen.add(LabelMen);
35
36         Menub1 = new JButton("BUY ITEMS");
37         Menub1.setBounds(200,80,300,45);
38         Menub1.addActionListener(this);
39         panelMen.add(Menub1);
40
41         /*Menub2 = new JButton("CART");
42         Menub2.setBounds(200,150,300,45);
43         panelMen.add(Menub2); */
44
45         JButton img = new JButton(new ImageIcon("C:\\Users\\TRENT\\Desktop\\College Works\\Advnced Prog\\resizeshef2.jpeg"));
46         img.setBounds(250, 150 , 180, 180);
47         panelMen.add(img);
48     }
49
50
51     public static void main (String[] args) {
52
53         Menu1 a = new Menu1();
54     }
55
56     @Override
57     public void actionPerformed(ActionEvent e) {
58         Details.main(null);
59     }
60 }
61
62
63
```

## OUTPUT:



## CUSTOMER DETAILS CODE:

```
1 import java.awt.Color;
2 import java.awt.Font;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 import javax.swing.*;
7
8 public class Details implements ActionListener {
9     private static JLabel label1;
10    private static JLabel label2;
11    private static JLabel label3;
12    private static JLabel label4;
13    private static JButton enter;
14    private static JTextField text1;
15    private static JTextField text2;
16    private static JTextField text3;
17
18    public Details(){
19        JPanel Panelc = new JPanel();
20        JFrame Framec = new JFrame();
21
22        Framec.setSize(700,500);
23        Framec.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24        Framec.setTitle("Customer Details");
25        Framec.setVisible(true);
26        Framec.setLocationRelativeTo(null);
27        Framec.setResizable(true);
28        Framec.add(Panelc);
29        Panelc.setLayout(null);
30        Panelc.setBackground(new Color(253,238,220));
31
32        label1 = new JLabel("Customer Details");
33        label1.setBounds(200,10,700,50);
34        label1.setFont(new Font("Arial", Font.BOLD, 30));
35        label1.setForeground(new Color(241, 161, 97));
36        Panelc.add(label1);
37
38        label2 = new JLabel("Customer Name:");
39        label2.setBounds(20,55,300,45);
40
41        label2 = new JLabel("Customer Name:");
42        label2.setBounds(20,55,300,45);
43        label2.setFont(new Font("Arial", Font.BOLD, 16));
44        Panelc.add(label2);
45
46        text1 = new JTextField();
47        text1.setBounds(180,65,260,25);
48        Panelc.add(text1);
49
50        label3 = new JLabel("Address:");
51        label3.setBounds(20,100,300,45);
52        label3.setFont(new Font("Arial", Font.BOLD, 16));
53        Panelc.add(label3);
54
55        text2 = new JTextField();
56        text2.setBounds(180,110,260,25);
57        Panelc.add(text2);
58
59        label4 = new JLabel("Contact No.:");
60        label4.setBounds(20,150,300,45);
61        label4.setFont(new Font("Arial", Font.BOLD, 16));
62        Panelc.add(label4);
63
64        text3 = new JTextField();
65        text3.setBounds(180,160,260,25);
66        Panelc.add(text3);
67
68        enter = new JButton("Enter");
69        enter.setBounds(190,210,300,45);
70        enter.addActionListener(this);
71        Panelc.add(enter);
72
73    }
74
75    public static void main(String[] args) {
76        Details i = new Details();
77    }
```



```

67     enter.addActionListener(this);
68     Panelc.add(enter);
69
70 }
71
72
73
74 public static void main(String[] args) {
75     Details i = new Details();
76
77
78 }
79
80
81 @Override
82 public void actionPerformed(ActionEvent e) {
83     String detail1 = text1.getText();
84     String detail2 = text2.getText();
85     String detail3 = text3.getText();
86
87     if (detail1.equals("Customer") && detail2.equals("123 Manila Street") && detail3.equals("1234567")){
88         JOptionPane.showMessageDialog(null, "Proceed to Shop");
89         BookStore.main(null);
90
91     }else {
92         JOptionPane.showMessageDialog(null, "Try Again");
93         text1.setText("");
94         text2.setText("");
95         text3.setText("");
96
97     }
98
99 }
100
101 }
102
103
104
105

```

OUTPUT:

