daniel2310990910@gmail.com

This document is conducted to help the understanding of ICT167 Assignment1 code which made by LEE JIHU.

ICT167 Assignmet1

[Date: Thursday, 4:15 p.m.] [LEE JIHU(CT0335241)]

1. **Title: Document for ICT167 assignment 1**

Author: LEE JIHU

Date: 20 June

File names: ICT167 ASSIGNMENT1

Purpose: This document is conducted to help the understanding of ICT Assignment1 code.

1. **Requirements/Specification**
2. Change class

In this class, the program will get the coin change amount and name of the owner of the coin.

1. getChange

In this method, the program will calculate and print the number of coins following the denominations (50, 20, 10, 5) of it.

1. Client class
2. data

In this method, the program will store the values of the names and coins which are prompt by user to array.

1. menu1

In this method, the program will ask the user to enter a name of the coin owner. Then search the name in the data and print the coin change amounts using the getChange method. If there is no name in the data, it will print an error message.

1. menu2

In this method, the program will print the name who has the largest amount of coins and the coin change amounts using the getChange method.

1. menu3

In this method, the program will print the name who has the smallest amount of coins and the coin change amounts using the getChange method.

1. menu4

In this method, the program will count the number of coins of all people in the data. And print the total number of coins.

1. main

In this method, the program will print the list of menus and prompt the menu number which user entered. The program will repeat this until user enters 5 (Exit).

1. User Guide
2. Enter 10 names and coin amounts.
3. The program will ask the user to select a menu. Enter an integer only.
4. Except menu1, the program dose not need anything from user.
5. In menu1, enter a name which is in the data. If not, it will print ‘NOT FOUND’.
6. The program will automatically repeat until user enters 5. Enter 5 to stop the program.

1. **Structure/Design/Algorithm**
2. Structure

I add an instance variable ‘changeArray’ to store the values in it and can be accessed from methods when those needs data.

1. Class diagram

**Change**

Name

coinChangeAmount

getChange

**client**

changeArray

data

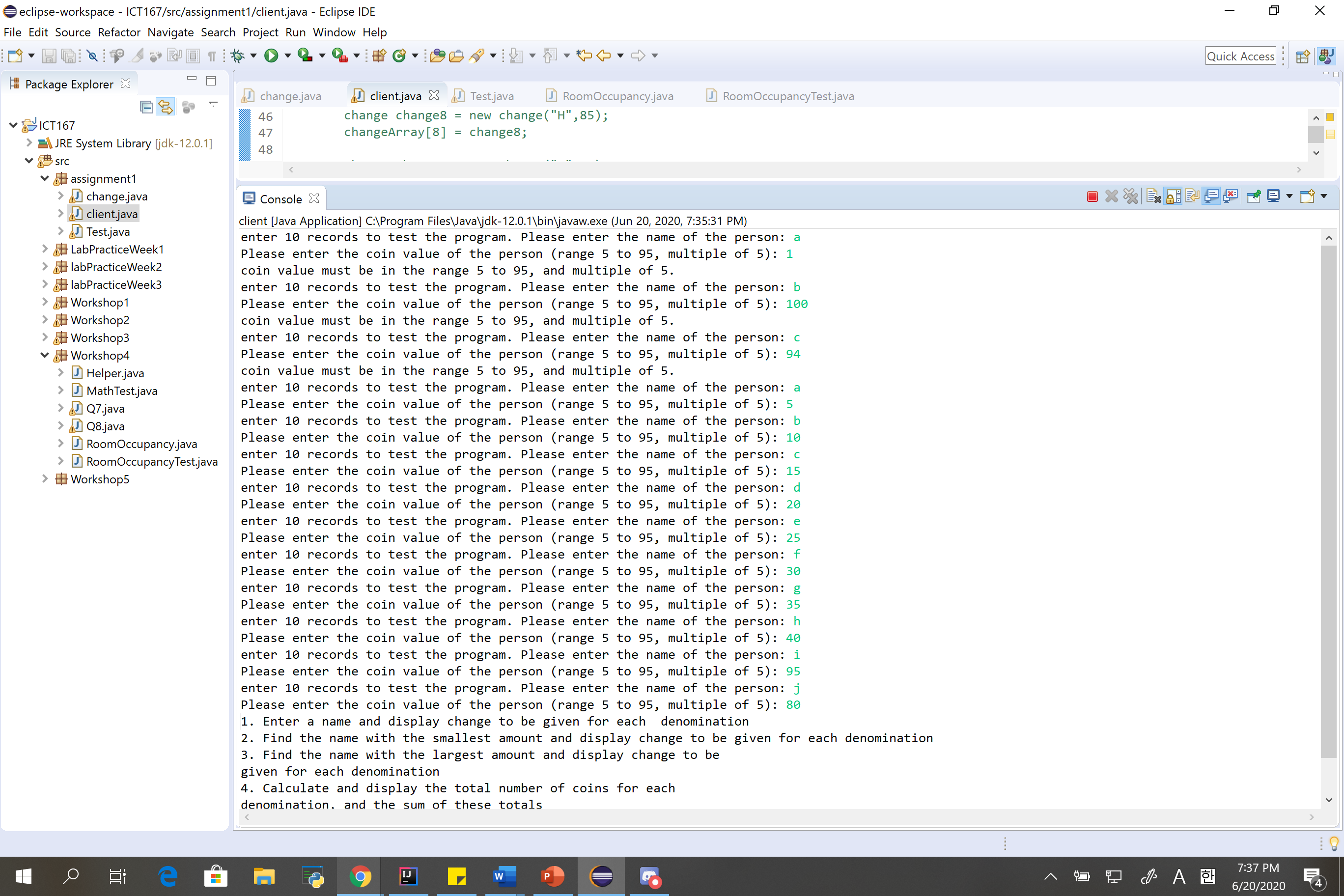
Use

Get data

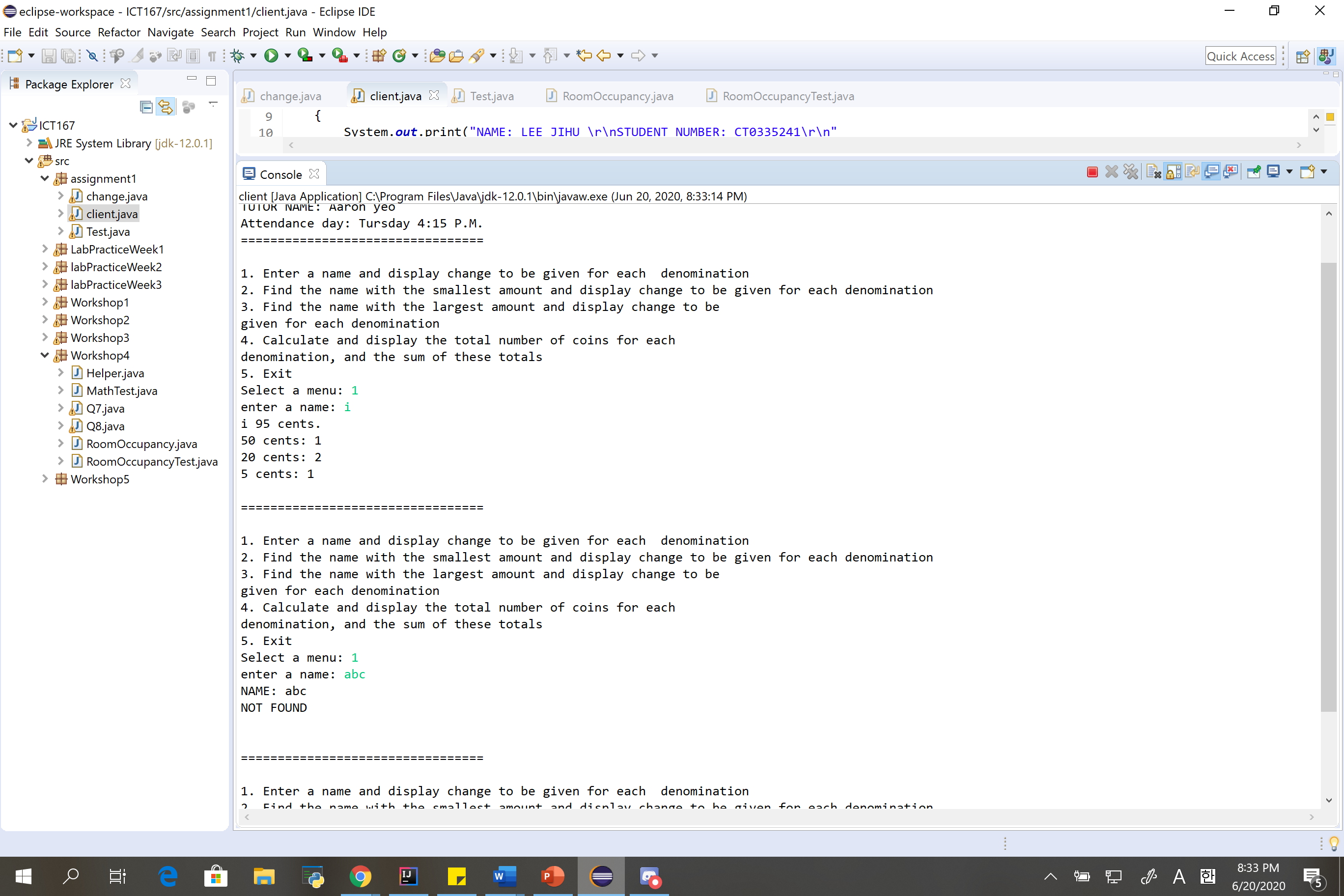
1. Description

Change class only has two instance variables and get change method which is the main functional part of the program. All extra programs like storing data or selecting menu is in the client class.

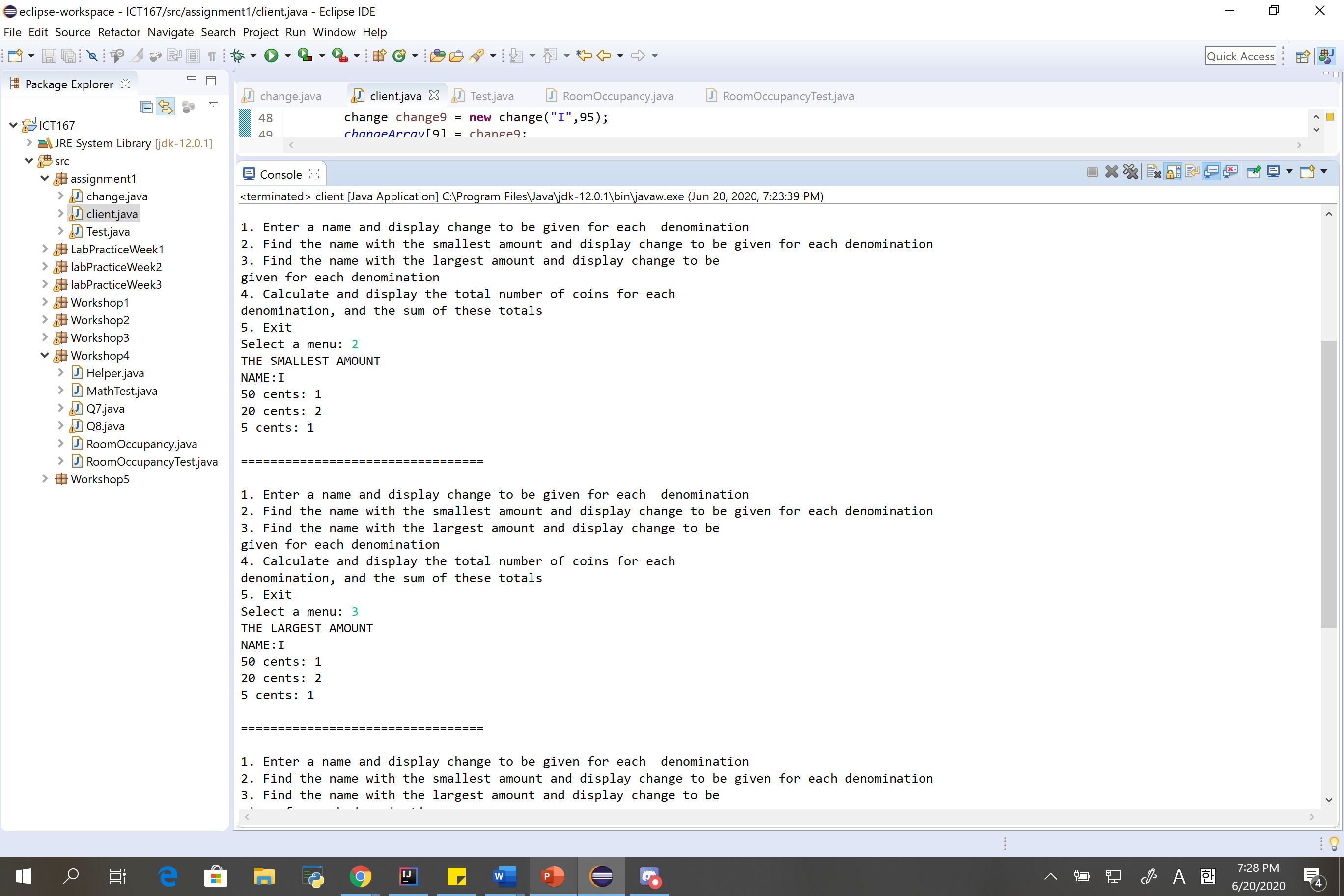
1. **Limitations**
2. There is a certain problem in the data (object, input). It automatically converts every value of the whole array to the last input. I could not figure out what to do in the data. The other codes were created assuming that the data was correct and has no error.
3. The program does not accept the input which has different type from the requiring.
4. **Testing**
5. getting data from use input

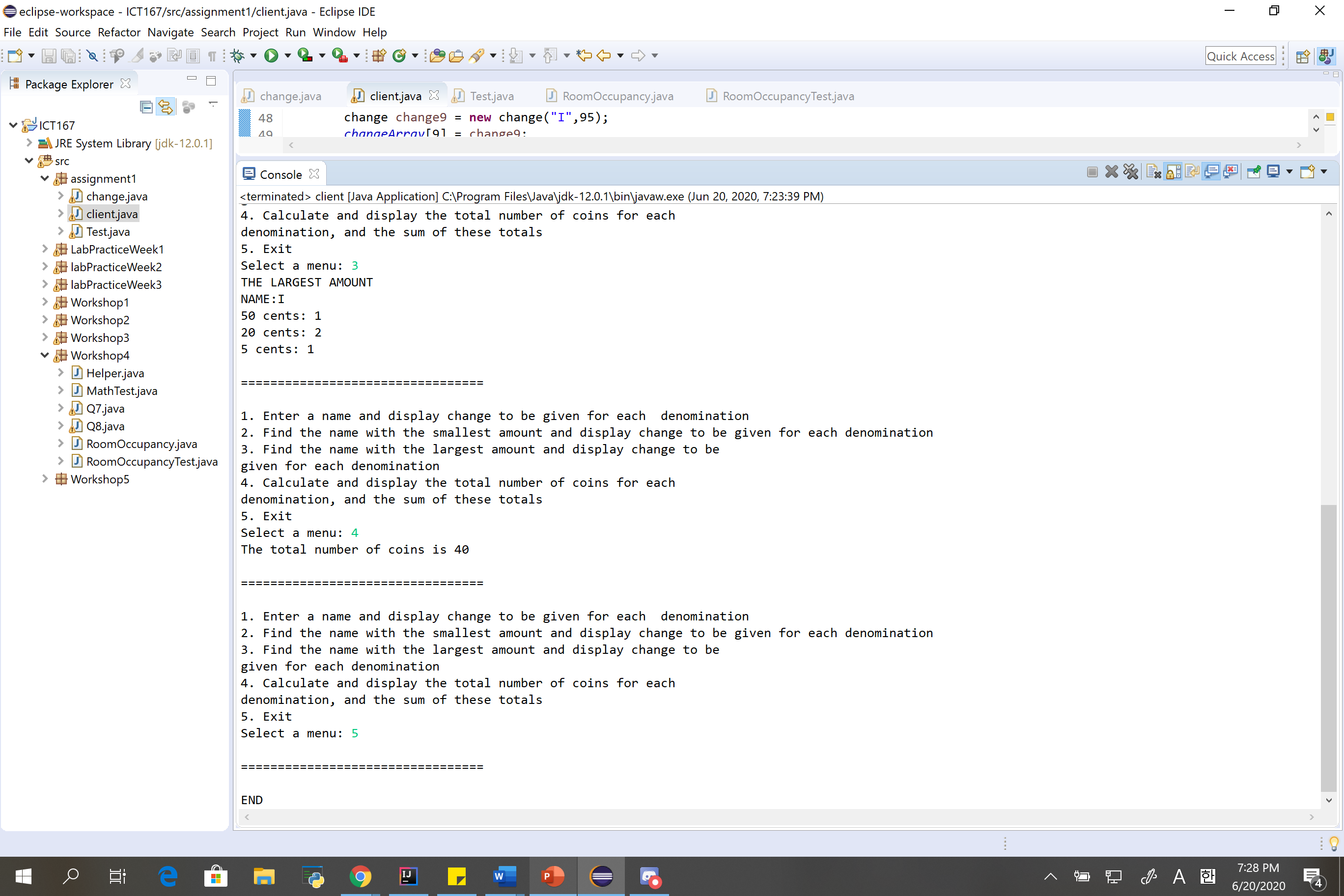


1. validating the input of menu1



1. menu2,3,4,5





1. **Java code**

**Client class**

1. **package** assignment1;
2. **import** java.util.Scanner;
3. **public** **class** client
4. {
6. **public** **static** **void** StudentInfo()
7. {
8. System.***out***.print("NAME: LEE JIHU \r\nSTUDENT NUMBER: CT0335241\r\n"
9. + "ENROLMENT: FTA ICT167\r\n"
10. + "TUTOR NAME: Aaron yeo\r\n"
11. + "Attendance day: Tursday 4:15 P.M.");
12. System.***out***.println("\r\n=================================\r\n");
13. }

16. **public** **static** change[] *changeArray* = **new** change[10];
18. /\*There is a certain problem in the data(object,input).
19. \* it automatically converts every value of the whole array to the last input.
20. \* I coudln't figure out what to do in the data.
21. \* The other codes were created assuming that the data was correct and has no error.
22. \*/
24. **public** **static** **void** data()
25. {
26. //purpose: storing the value changeArray.
28. //store into array

31. change change0 = **new** change("John",5);
32. *changeArray*[0] = change0;
34. change change1 = **new** change("A",15);
35. *changeArray*[1] = change1;
36. change change2 = **new** change("B",25);
37. *changeArray*[2] = change2;
39. change change3 = **new** change("C",35);
40. *changeArray*[3] = change3;
42. change change4 = **new** change("D",45);
43. *changeArray*[4] = change4;
45. change change5 = **new** change("E",55);
46. *changeArray*[5] = change5;
48. change change6 = **new** change("F",65);
49. *changeArray*[6] = change6;
51. change change7 = **new** change("G",75);
52. *changeArray*[7] = change7;
53. change change8 = **new** change("H",85);
54. *changeArray*[8] = change8;
56. change change9 = **new** change("I",95);
57. *changeArray*[9] = change9;

60. /\*
61. \* //getting data from user.
63. Scanner kb = new Scanner(System.in);

66. for (int i = 0; i < 10; i++)
68. {
69. System.out.print("enter 10 records to test the program. Please enter the name of the person: ");
70. String name = kb.next();
71. System.out.print("Please enter the coin value of the person (range 5 to 95, multiple of 5): ");
72. int coin = kb.nextInt();
73. if ((coin % 5 == 0) && (coin <= 95) && (coin >= 5))
74. {
75. changeArray[i].name = name;
76. changeArray[i].coinChangeAmount = coin;
77. }
78. else
79. {
80. System.out.println("coin value must be in the range 5 to 95, and multiple of 5. ");
81. i--;
82. }
83. }

86. // if same name, add coin.
87. for(int count=0; count<10; count++)
88. {
89. for(int i=0; i<10; i++)
90. {
91. if (changeArray[count].name==changeArray[i].name)
92. {
93. changeArray[count].coinChangeAmount += changeArray[i].coinChangeAmount;
94. }
95. }
96. count++;
97. }
98. \*/
100. }

103. **public** **static** **void** menu1()
104. {
106. Scanner kb = **new** Scanner(System.***in***);
107. System.***out***.print("enter a name: ");
108. String Customer = kb.nextLine();
109. **int** n=0;
110. **boolean** flag=**false**;
112. **for** (**int** i=0;i<10;i++)
113. {
114. **if** (Customer.equalsIgnoreCase(*changeArray*[i].*name*))
115. {
116. flag=**true**;
117. n=i;
118. }
119. }
121. **if**(flag)
122. {
123. System.***out***.println(Customer+" "+*changeArray*[n].*coinChangeAmount*+" cents.");
124. change.*getChange*(*changeArray*[n].*coinChangeAmount*);
125. }
126. **else**
127. {
128. System.***out***.println("NAME: "+Customer+"\r\nNOT FOUND\r\n");
129. }
130. System.***out***.println("\r\n=================================\r\n");
131. }

134. **public** **static** **void** menu2()
135. {
137. **int** min=0;
138. **int** n=0;
140. **for** (**int** i=0;i<10;i++)
141. {
142. **if** (i==0)
143. {
144. min=*changeArray*[i].*coinChangeAmount*;
145. n=i;
146. }
147. **else**
148. {
149. **if** (min>*changeArray*[i].*coinChangeAmount*)
150. {
151. min=*changeArray*[i].*coinChangeAmount*;
152. n=i;
153. }
154. }
155. }
156. System.***out***.println("THE SMALLEST AMOUNT \r\nNAME:"+*changeArray*[n].*name*);
157. change.*getChange*(*changeArray*[n].*coinChangeAmount*);
158. System.***out***.println("\r\n=================================\r\n");
160. }
162. **public** **static** **void** menu3()
163. {
165. **int** max=0;
166. **int** n=0;
168. **for** (**int** i=0;i<10;i++)
169. {
170. **if** (i==0)
171. {
172. max=*changeArray*[i].*coinChangeAmount*;
173. n=i;
174. }
175. **else**
176. {
177. **if** (max<*changeArray*[i].*coinChangeAmount*)
178. {
179. max=*changeArray*[i].*coinChangeAmount*;
180. n=i;
181. }
182. }
183. }
184. System.***out***.println("THE LARGEST AMOUNT\r\nNAME:"+*changeArray*[n].*name*);
185. change.*getChange*(*changeArray*[n].*coinChangeAmount*);
186. System.***out***.println("\r\n=================================\r\n");
187. }
189. **public** **static** **void** menu4()
190. {
192. **int**[] changeToReturn = **new** **int**[4];
193. **int**[] coin = {50, 20, 10, 5};
194. **int** count1=0, count2=0;
196. **for** (**int** i0=0; i0<10; i0++)
197. {
198. **for**(**int** i=0; i<coin.length; i++)
199. {
200. changeToReturn[i]=*changeArray*[i].*coinChangeAmount*/coin[i];
201. *changeArray*[i].*coinChangeAmount* -= coin[i]\*changeToReturn[i];
202. count1+=changeToReturn[i];
203. }
204. count2+=count1;
205. }
207. System.***out***.println("The total number of coins is "+ count2);
208. System.***out***.println("\r\n=================================\r\n");
210. }

213. **public** **static** **void** main(String[] args)
214. {
215. *StudentInfo*();
216. *data*();
218. //menu
219. Scanner kb = **new** Scanner(System.***in***);
221. **int** menu=0;
223. **while**(menu != 5)
224. {
226. System.***out***.print(
227. "1. Enter a name and display change to be given for each denomination\r\n" +
228. "2. Find the name with the smallest amount and display change to be given for each denomination\r\n" +
229. "3. Find the name with the largest amount and display change to be\r\n" +
230. "given for each denomination\r\n" +
231. "4. Calculate and display the total number of coins for each\r\n" +
232. "denomination, and the sum of these totals\r\n" +
233. "5. Exit\r\n" +
234. "Select a menu: ");
235. menu = kb.nextInt();
237. **switch**(menu)
238. {
239. **case** 1:
240. *menu1*();
241. **break**;
243. **case** 2:
244. *menu2*();
245. **break**;
247. **case** 3:
248. *menu3*();
249. **break**;
251. **case** 4:
252. *menu4*();
253. **break**;
255. **default**:
256. **if** (menu!=5)
257. {
258. System.***out***.println("please enter a number between 1 to 5.");
259. }
260. System.***out***.println("\r\n=================================\r\n");
261. **break**;
262. }
264. }
266. System.***out***.print("END");
268. }
269. }

**Change class**

1. **package** assignment1;
2. **public** **class** change
3. {
4. //50, 20, 10 and 5 cents
5. **public** **static** String *name*;
6. **public** **static** **int** *coinChangeAmount*;

9. **public** change(String name, **int** coinChangeAmount)
10. {
11. **this**.*name* = name;
12. **this**.*coinChangeAmount* = coinChangeAmount;
13. }

16. **public** **static** String getName() {
17. **return** *name*;
18. }
19. **public** **static** **void** setName(String name) {
20. change.*name* = name;
21. }
22. **public** **static** **int** getCoinChangeAmount() {
23. **return** *coinChangeAmount*;
24. }
25. **public** **static** **void** setCoinChangeAmount(**int** coinChangeAmount) {
26. change.*coinChangeAmount* = coinChangeAmount;
27. }
28. **public** **static** **void** getChange(**int** CA)
29. {
30. **int**[] changeToReturn = **new** **int**[4];
31. **int**[] coin = {50, 20, 10, 5};
33. **for**(**int** i=0; i<coin.length; i++)
34. {
35. changeToReturn[i]=CA/coin[i];
36. CA -= coin[i]\*changeToReturn[i];
38. **if**(changeToReturn[i]!=0)
39. {
40. **if** (i==0)
41. {
42. System.***out***.println("50 cents: "+changeToReturn[i]);
43. }
44. **else** **if** (i==1)
45. {
46. System.***out***.println("20 cents: "+changeToReturn[i]);
47. }
48. **else** **if** (i==2)
49. {
50. System.***out***.println("10 cents: "+changeToReturn[i]);
51. }
52. **else** **if** (i==3)
53. {
54. System.***out***.println("5 cents: "+changeToReturn[i]);
55. }
56. }
57. }
58. }
60. }