**Assignment2Main.java**

1 public class Assignment2Main **{**

2 public static void main**(**String**[]** args**)** **{**

3

4 CreateCust obj1 **=** **new** CreateCust**();** //creating customer object

5

6 obj1**.**CreateCustomer**();** //calling customer object

7

8 **}**

9 **}**

**checkOUT.java**

1 **import** java**.**util**.\*;**

2 // thread to checkout customers in queue

3 class checkOUT **implements** Runnable**{**

4 int time**;**

5 Random r **=** **new** Random**();**

6 Queue**<**Integer**>** q **=** **new** LinkedList**<**Integer**>();**//creating local queue

7

8 public checkOUT**(**Queue**<**Integer**>** x**){**//passing queue into thread

9 q **=** x**;** //inputing our queue into thread queue

10 **}**

11

12 public void run**(){**

13 **try{**

14 Thread**.**sleep**(**1000**);**// waiting 1 second before removing queue

15 **for(**int num **=** 1**;** num **<=** 5000**;** num**++){**// removes from queue one at time

16 time **=** q**.**peek**();**

17 int number **=** time **\*** 10**;**

18 Thread**.**sleep**(**number**);**

19 q**.**remove**();**

20 **}**

21 **}catch(**Exception e**){**

22 System**.**out**.**println**(**"Queue is Empty"**);** //lets user know queue is empty

23 **}**

24 **}**

25 **}**

**CreateCust.java**

1 **import** java**.**util**.**Random**;**

2 public class CreateCust **implements** Runnable**{**

3 Random myRND **=** **new** Random**();** //random variable creation

4 Random rnd **=** **new** Random**();** // random variable creation

5 int someNUM **=** 1**;** // used for simply seeing the counter number

6 int maxSize **=** 1000**;** // array size

7 Functions arr **=** **new** Functions**(**maxSize**);** // reference to and creation of array

8

9 public void run**(){** // following is timer thread

10 **try{**

11 **for** **(**int num **=** 1**;** num **<=** 70**;** num**++){**

12 System**.**out**.**println**(**"Current time: " **+** num**);**

13 Thread**.**sleep**(**1000**);**

14 **}**

15 **}catch(**Exception e**){}**

16 **}**

17

18 public void CreateCustomer**(){**

19 Thread timer **=** **new** Thread**(new** CreateCust**());**

20

21 **try{**

22 System**.**out**.**println**(**"Creating Customers and their Items. Will show Shortly. \n"

23 **+** "\*note\* Customer is being generated at 1-10 ticks randomly \n"**);**

24 System**.**out**.**println**(**"Customer Number | Number of Items"**);**

25 System**.**out**.**println**(**"---------------------------------"**);**

26 // for loop will loop multiple times to fill customer items and create customer

27 **for(**int custNUM **=** 1**;** custNUM **<=** 1000**;** custNUM**++){**

28 int amt **=** rnd**.**nextInt**(**10**)** **+** 1**;** // will generate a random number between 1-10

29

30 Thread**.**sleep**(**amt**);** // will sleep until number items (milliseconds)

31

32 int amtITEMS **=** myRND**.**nextInt**(**100**)** **+** 1**;** // will generate a random number between 1-100 that will be used as customer items

33 arr**.**insert**(**custNUM**,** amtITEMS**);** // customer number will be inserted and random amount of items will be inserted.

34 arr**.**displayA**();**

35 **}**

36 **}catch(**Exception e**){};**

37 timer**.**start**();**

38 System**.**out**.**println**(**"Now the customers are being stored into the Chasiers and being Checked out. \n"

39 **+** "note\* queue empty message will display 10 times when all 10 chasiers are done \n"

40 **+** "Thank You!"**);**

41 arr**.**numItems**();**

42 **}**

43 **}**

**Customer.java**

1 public class Customer **{**

2 private int nCustomer**;**

3 private int nItems**;**

4 //--------------------------------------------------------------

5 public Customer**(**int cust**,** int item**){** // constructor

6 nCustomer **=** cust**;**

7 nItems **=** item**;**

8 **}**

9 public void displayCustomer**(){**

10 System**.**out**.**println**(**" " **+** nCustomer **+** " | " **+** nItems **+** " "**);**

11 **}**

12 public int getItems**(){** // get number of items

13 **return** nItems**;**

14 **}**

15 public int getCustomer**(){** // get last name

16 **return** nCustomer**;**

17 **}**

18 **}**

**Functions.java without Express Lines**

1 **import** java**.**util**.**LinkedList**;**

2 **import** java**.**util**.**Queue**;**

3 public class Functions **{**

4 int nPosition**;**

5 int j **=** 0**;**

6

7 // create 10 Chasiers

8 Queue**<**Integer**>** myCSH0 **=** **new** LinkedList**<**Integer**>();**

9 Queue**<**Integer**>** myCSH1 **=** **new** LinkedList**<**Integer**>();**

10 Queue**<**Integer**>** myCSH2 **=** **new** LinkedList**<**Integer**>();**

11 Queue**<**Integer**>** myCSH3 **=** **new** LinkedList**<**Integer**>();**

12 Queue**<**Integer**>** myCSH4 **=** **new** LinkedList**<**Integer**>();**

13 Queue**<**Integer**>** myCSH5 **=** **new** LinkedList**<**Integer**>();**

14 Queue**<**Integer**>** myCSH6 **=** **new** LinkedList**<**Integer**>();**

15 Queue**<**Integer**>** myCSH7 **=** **new** LinkedList**<**Integer**>();**

16 Queue**<**Integer**>** myCSH8 **=** **new** LinkedList**<**Integer**>();**

17 Queue**<**Integer**>** myCSH9 **=** **new** LinkedList**<**Integer**>();**

18

19 private Customer**[]** a**;** // reference to array

20 private int nElems**;** // number of data items

21

22 public Functions**(**int max**){** //constructor

23 a **=** **new** Customer**[**max**];** // create the array

24 nElems **=** 0**;** // no items yet

25 **}**

26 public void insert**(**int cust**,** int item**){** // put person into array

27 a**[**nElems**]** **=** **new** Customer**(**cust**,** item**);**

28 nElems**++;** // increment size

29 **}**

30 public void displayA**(){**

31 a**[**j**].**displayCustomer**();** // display it

32 j**++;**

33 **}**

34 // take the number of items by the customer and insert into a queue

35 public void numItems**(){**

36 nPosition **=** 0**;**

37

38 int**[]** tmp **=** **new** int**[**10**];**

39

40 tmp**[**0**]** **=** a**[**nPosition**].**getItems**();**

41 nPosition**++;**

42 tmp**[**1**]** **=** a**[**nPosition**].**getItems**();**

43 nPosition**++;**

44 tmp**[**2**]** **=** a**[**nPosition**].**getItems**();**

45 nPosition**++;**

46 tmp**[**3**]** **=** a**[**nPosition**].**getItems**();**

47 nPosition**++;**

48 tmp**[**4**]** **=** a**[**nPosition**].**getItems**();**

49 nPosition**++;**

50 tmp**[**5**]** **=** a**[**nPosition**].**getItems**();**

51 nPosition**++;**

52 tmp**[**6**]** **=** a**[**nPosition**].**getItems**();**

53 nPosition**++;**

54 tmp**[**7**]** **=** a**[**nPosition**].**getItems**();**

55 nPosition**++;**

56 tmp**[**8**]** **=** a**[**nPosition**].**getItems**();**

57 nPosition**++;**

58 tmp**[**9**]** **=** a**[**nPosition**].**getItems**();**

59 nPosition**++;**

60

61 myCSH0**.**add**(**tmp**[**0**]);** //Inserts into the Queue

62 myCSH1**.**add**(**tmp**[**1**]);**

63 myCSH2**.**add**(**tmp**[**2**]);**

64 myCSH3**.**add**(**tmp**[**3**]);**

65 myCSH4**.**add**(**tmp**[**4**]);**

66 myCSH5**.**add**(**tmp**[**5**]);**

67 myCSH6**.**add**(**tmp**[**6**]);**

68 myCSH7**.**add**(**tmp**[**7**]);**

69 myCSH8**.**add**(**tmp**[**8**]);**

70 myCSH9**.**add**(**tmp**[**9**]);**

71

72 //creating checkout thread object

73 Thread t0 **=** **new** Thread**(new** checkOUT**(**myCSH0**));**

74 Thread t1 **=** **new** Thread**(new** checkOUT**(**myCSH1**));**

75 Thread t2 **=** **new** Thread**(new** checkOUT**(**myCSH2**));**

76 Thread t3 **=** **new** Thread**(new** checkOUT**(**myCSH3**));**

77 Thread t4 **=** **new** Thread**(new** checkOUT**(**myCSH4**));**

78 Thread t5 **=** **new** Thread**(new** checkOUT**(**myCSH5**));**

79 Thread t6 **=** **new** Thread**(new** checkOUT**(**myCSH6**));**

80 Thread t7 **=** **new** Thread**(new** checkOUT**(**myCSH7**));**

81 Thread t8 **=** **new** Thread**(new** checkOUT**(**myCSH8**));**

82 Thread t9 **=** **new** Thread**(new** checkOUT**(**myCSH9**));**

83

84 //starting checkout thread

85 t0**.**start**();**

86 t1**.**start**();**

87 t2**.**start**();**

88 t3**.**start**();**

89 t4**.**start**();**

90 t5**.**start**();**

91 t6**.**start**();**

92 t7**.**start**();**

93 t8**.**start**();**

94 t9**.**start**();**

95

96 **try{**

97 //adding into chasier queue by comparing who is short

98 **for** **(**int num **=** 1**;** num **<=** 990**;** num**++){**

99 **if** **(**myCSH0**.**size**()** **<** myCSH9**.**size**()){**

100 tmp**[**0**]** **=** a**[**nPosition**].**getItems**();**

101 myCSH0**.**add**(**tmp**[**0**]);**

102 nPosition**++;**

103 **}**

104 **else** **if(**myCSH1**.**size**()** **<** myCSH0**.**size**()){**

105 tmp**[**1**]** **=** a**[**nPosition**].**getItems**();**

106 myCSH1**.**add**(**tmp**[**1**]);**

107 nPosition**++;**

108 **}**

109 **else** **if(**myCSH2**.**size**()** **<** myCSH1**.**size**()){**

110 tmp**[**2**]** **=** a**[**nPosition**].**getItems**();**

111 myCSH2**.**add**(**tmp**[**2**]);**

112 nPosition**++;**

113 **}**

114 **else** **if(**myCSH3**.**size**()** **<** myCSH2**.**size**()){**

115 tmp**[**3**]** **=** a**[**nPosition**].**getItems**();**

116 myCSH3**.**add**(**tmp**[**3**]);**

117 nPosition**++;**

118 **}**

119 **else** **if(**myCSH4**.**size**()** **<** myCSH3**.**size**()){**

120 tmp**[**4**]** **=** a**[**nPosition**].**getItems**();**

121 myCSH4**.**add**(**tmp**[**4**]);**

122 nPosition**++;**

123 **}**

124 **else** **if(**myCSH5**.**size**()** **<** myCSH4**.**size**()){**

125 tmp**[**5**]** **=** a**[**nPosition**].**getItems**();**

126 myCSH5**.**add**(**tmp**[**5**]);**

127 nPosition**++;**

128 **}**

129 **else** **if(**myCSH6**.**size**()** **<** myCSH5**.**size**()){**

130 tmp**[**6**]** **=** a**[**nPosition**].**getItems**();**

131 myCSH6**.**add**(**tmp**[**6**]);**

132 nPosition**++;**

133 **}**

134 **else** **if(**myCSH7**.**size**()** **<** myCSH6**.**size**()){**

135 tmp**[**7**]** **=** a**[**nPosition**].**getItems**();**

136 myCSH7**.**add**(**tmp**[**7**]);**

137 nPosition**++;**

138 **}**

139 **else** **if(**myCSH8**.**size**()** **<** myCSH7**.**size**()){**

140 tmp**[**8**]** **=** a**[**nPosition**].**getItems**();**

141 myCSH8**.**add**(**tmp**[**8**]);**

142 nPosition**++;**

143 **}**

144 **else** **if(**myCSH9**.**size**()** **<** myCSH8**.**size**()){**

145 tmp**[**9**]** **=** a**[**nPosition**].**getItems**();**

146 myCSH9**.**add**(**tmp**[**9**]);**

147 nPosition**++;**

148 **}**

149 **else{**

150 tmp**[**0**]** **=** a**[**nPosition**].**getItems**();**

151 myCSH0**.**add**(**tmp**[**0**]);**

152 nPosition**++;**

153 **}**

154 Thread**.**sleep**(**10**);**//sleeps for 10 milli seconds

155 **}**

156 **}catch(**Exception e**){**

157 System**.**out**.**println**(**"OVERFLOW, but its okay."**);**

158 **}**// displays when loop overflows

159 **}**

160 **}**

**Functions.java with Express Lines**

1 **import** java**.**util**.**LinkedList**;**

2 **import** java**.**util**.**Queue**;**

3 public class Functions **{**

4 int nPosition**;**

5 int j **=** 0**;**

6

7 // create 10 Chasiers

8 Queue**<**Integer**>** myCSH0 **=** **new** LinkedList**<**Integer**>();**

9 Queue**<**Integer**>** myCSH1 **=** **new** LinkedList**<**Integer**>();**

10 Queue**<**Integer**>** myCSH2 **=** **new** LinkedList**<**Integer**>();**

11 Queue**<**Integer**>** myCSH3 **=** **new** LinkedList**<**Integer**>();**

12 Queue**<**Integer**>** myCSH4 **=** **new** LinkedList**<**Integer**>();**

13 Queue**<**Integer**>** myCSH5 **=** **new** LinkedList**<**Integer**>();**

14 Queue**<**Integer**>** myCSH6 **=** **new** LinkedList**<**Integer**>();**

15 Queue**<**Integer**>** myCSH7 **=** **new** LinkedList**<**Integer**>();**

16 Queue**<**Integer**>** myCSH8 **=** **new** LinkedList**<**Integer**>();**

17 Queue**<**Integer**>** myCSH9 **=** **new** LinkedList**<**Integer**>();**

18

19 private Customer**[]** a**;** // reference to array

20 private int nElems**;** // number of data items

21

22 public Functions**(**int max**){** //constructor

23 a **=** **new** Customer**[**max**];** // create the array

24 nElems **=** 0**;** // no items yet

25 **}**

26 public void insert**(**int cust**,** int item**){** // put person into array

27 a**[**nElems**]** **=** **new** Customer**(**cust**,** item**);**

28 nElems**++;** // increment size

29 **}**

30 public void displayA**(){**

31 a**[**j**].**displayCustomer**();** // display it

32 j**++;**

33 **}**

34 // take the number of items by the customer and insert into a queue

35 public void numItems**(){**

36 nPosition **=** 0**;**

37

38 int**[]** tmp **=** **new** int**[**10**];**

39

40 tmp**[**0**]** **=** a**[**nPosition**].**getItems**();**

41 nPosition**++;**

42 tmp**[**1**]** **=** a**[**nPosition**].**getItems**();**

43 nPosition**++;**

44 tmp**[**2**]** **=** a**[**nPosition**].**getItems**();**

45 nPosition**++;**

46 tmp**[**3**]** **=** a**[**nPosition**].**getItems**();**

47 nPosition**++;**

48 tmp**[**4**]** **=** a**[**nPosition**].**getItems**();**

49 nPosition**++;**

50 tmp**[**5**]** **=** a**[**nPosition**].**getItems**();**

51 nPosition**++;**

52 tmp**[**6**]** **=** a**[**nPosition**].**getItems**();**

53 nPosition**++;**

54 tmp**[**7**]** **=** a**[**nPosition**].**getItems**();**

55 nPosition**++;**

56 tmp**[**8**]** **=** a**[**nPosition**].**getItems**();**

57 nPosition**++;**

58 tmp**[**9**]** **=** a**[**nPosition**].**getItems**();**

59 nPosition**++;**

60

61 myCSH0**.**add**(**tmp**[**0**]);** //Inserts into the Queue

62 myCSH1**.**add**(**tmp**[**1**]);**

63 myCSH2**.**add**(**tmp**[**2**]);**

64 myCSH3**.**add**(**tmp**[**3**]);**

65 myCSH4**.**add**(**tmp**[**4**]);**

66 myCSH5**.**add**(**tmp**[**5**]);**

67 myCSH6**.**add**(**tmp**[**6**]);**

68 myCSH7**.**add**(**tmp**[**7**]);**

69 myCSH8**.**add**(**tmp**[**8**]);**

70 myCSH9**.**add**(**tmp**[**9**]);**

71

72 //creating checkout thread object

73 Thread t0 **=** **new** Thread**(new** checkOUT**(**myCSH0**));**

74 Thread t1 **=** **new** Thread**(new** checkOUT**(**myCSH1**));**

75 Thread t2 **=** **new** Thread**(new** checkOUT**(**myCSH2**));**

76 Thread t3 **=** **new** Thread**(new** checkOUT**(**myCSH3**));**

77 Thread t4 **=** **new** Thread**(new** checkOUT**(**myCSH4**));**

78 Thread t5 **=** **new** Thread**(new** checkOUT**(**myCSH5**));**

79 Thread t6 **=** **new** Thread**(new** checkOUT**(**myCSH6**));**

80 Thread t7 **=** **new** Thread**(new** checkOUT**(**myCSH7**));**

81 Thread t8 **=** **new** Thread**(new** checkOUT**(**myCSH8**));**

82 Thread t9 **=** **new** Thread**(new** checkOUT**(**myCSH9**));**

83

84 //starting checkout thread

85 t0**.**start**();**

86 t1**.**start**();**

87 t2**.**start**();**

88 t3**.**start**();**

89 t4**.**start**();**

90 t5**.**start**();**

91 t6**.**start**();**

92 t7**.**start**();**

93 t8**.**start**();**

94 t9**.**start**();**

95

96 **try{**

97 //adding into chasier queue by comparing who is short

98 **for** **(**int num **=** 1**;** num **<=** 990**;** num**++){**

99 int number20 **=** a**[**nPosition**].**getItems**();**

100 //checks if less than 20 items

101 **if** **(**number20 **<=** 20**){**

102 **if** **(**myCSH0**.**size**()** **<** myCSH1**.**size**()){**

103 tmp**[**0**]** **=** a**[**nPosition**].**getItems**();**

104 myCSH0**.**add**(**tmp**[**0**]);**

105 nPosition**++;**

106 **}**

107 **else** **if(**myCSH1**.**size**()** **<** myCSH0**.**size**()){**

108 tmp**[**1**]** **=** a**[**nPosition**].**getItems**();**

109 myCSH1**.**add**(**tmp**[**1**]);**

110 nPosition**++;**

111 **}**

112 **else{**

113 tmp**[**0**]** **=** a**[**nPosition**].**getItems**();**

114 myCSH0**.**add**(**tmp**[**0**]);**

115 nPosition**++;**

116 **}**

117 **}**

118 **else{**

119 **if(**myCSH2**.**size**()** **<** myCSH9**.**size**()){**

120 tmp**[**2**]** **=** a**[**nPosition**].**getItems**();**

121 myCSH2**.**add**(**tmp**[**2**]);**

122 nPosition**++;**

123 **}**

124 **else** **if(**myCSH3**.**size**()** **<** myCSH2**.**size**()){**

125 tmp**[**3**]** **=** a**[**nPosition**].**getItems**();**

126 myCSH3**.**add**(**tmp**[**3**]);**

127 nPosition**++;**

128 **}**

129 **else** **if(**myCSH4**.**size**()** **<** myCSH3**.**size**()){**

130 tmp**[**4**]** **=** a**[**nPosition**].**getItems**();**

131 myCSH4**.**add**(**tmp**[**4**]);**

132 nPosition**++;**

133 **}**

134 **else** **if(**myCSH5**.**size**()** **<** myCSH4**.**size**()){**

135 tmp**[**5**]** **=** a**[**nPosition**].**getItems**();**

136 myCSH5**.**add**(**tmp**[**5**]);**

137 nPosition**++;**

138 **}**

139 **else** **if(**myCSH6**.**size**()** **<** myCSH5**.**size**()){**

140 tmp**[**6**]** **=** a**[**nPosition**].**getItems**();**

141 myCSH6**.**add**(**tmp**[**6**]);**

142 nPosition**++;**

143 **}**

144 **else** **if(**myCSH7**.**size**()** **<** myCSH6**.**size**()){**

145 tmp**[**7**]** **=** a**[**nPosition**].**getItems**();**

146 myCSH7**.**add**(**tmp**[**7**]);**

147 nPosition**++;**

148 **}**

149 **else** **if(**myCSH8**.**size**()** **<** myCSH7**.**size**()){**

150 tmp**[**8**]** **=** a**[**nPosition**].**getItems**();**

151 myCSH8**.**add**(**tmp**[**8**]);**

152 nPosition**++;**

153 **}**

154 **else** **if(**myCSH9**.**size**()** **<** myCSH8**.**size**()){**

155 tmp**[**9**]** **=** a**[**nPosition**].**getItems**();**

156 myCSH9**.**add**(**tmp**[**9**]);**

157 nPosition**++;**

158 **}**

159 **else{**

160 tmp**[**2**]** **=** a**[**nPosition**].**getItems**();**

161 myCSH2**.**add**(**tmp**[**2**]);**

162 nPosition**++;**

163 **}**

164 **}**

165 Thread**.**sleep**(**10**);**//sleeps for 10 milli seconds

166 **}**

167 **}catch(**Exception e**){**

168 System**.**out**.**println**(**"OVERFLOW, but its okay."**);**

169 **}**// displays when loop overflows

170 **}**

171 **}**