# **ICT167 Assignment 1**

Name: Ang Jin Wei

Date: 14 Feb 2023

Filenames: Changecoin.java, Client.java

Purpose: ICT167 Assignment 1 External Documentation

## **Project Requirements/Specifications**

The program will read the user’s input for names and coin values and store it in an array. The program will then prompt the user to pick from a choice of 1,2,3,4,5,6 and perform a variety of functions based on the choice chosen. The coin inputs will be calculated based on Australia’s coin denominations, from two dollars to five cents.

Example of denomination calculation:

Name: Mary

Coin: 200

Output:

2 dollars - 1

Choice 1 - The program will prompt the user to enter a name and display the amount of change given for each denomination for the particular person.

Choice 2 - The program will look for the person with the largest amount of coins and display the change given for each denomination.

Choice 3 - The program will look for the person with the smallest amount of coins and display the change given for each denomination.

Choice 4 - The program will calculate the total number of coins and display the change given for each denomination.

Choice 5 - The program will calculate the total amount value of coins and display the amount in cents for each denomination.

Choice 6 - The program will print a farewell message and exit the program.

## **User Guide**

The IDE used for this program is Apache Netbeans IDE 16.

Changecoin.java provides the class Changecoin with a constructor with two parameters.

Client.java is the client class that executes the methods found in the Changecoin class.

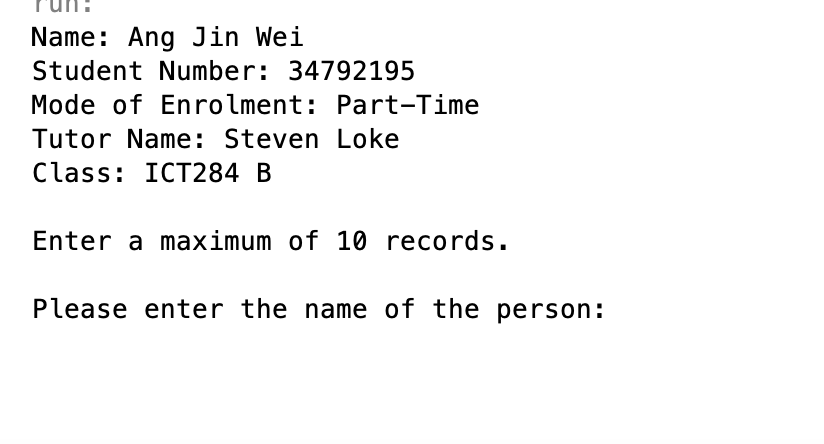
To run the program, follow the following steps:

Step 1 - Start your IDE

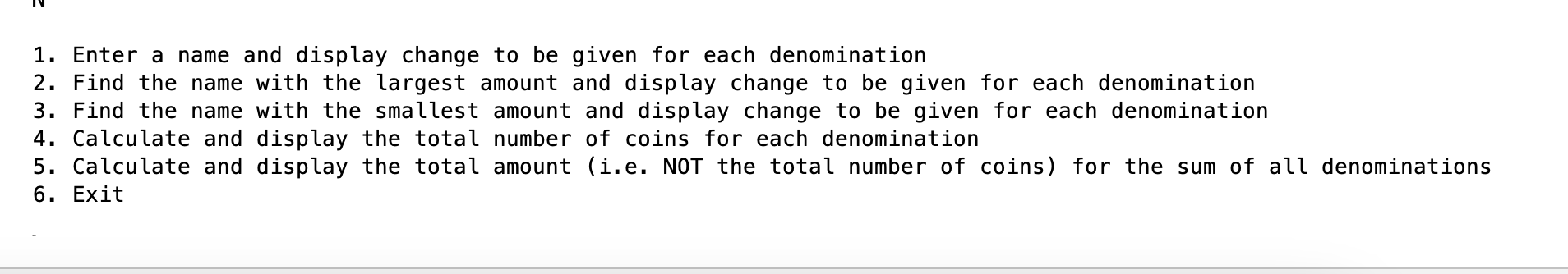
Step 2 - Open “Assignment1” source

Step 3 - Run the program and follow the prompts on the program

Step 4 -

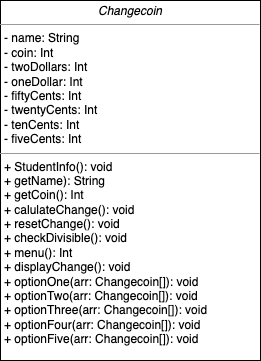
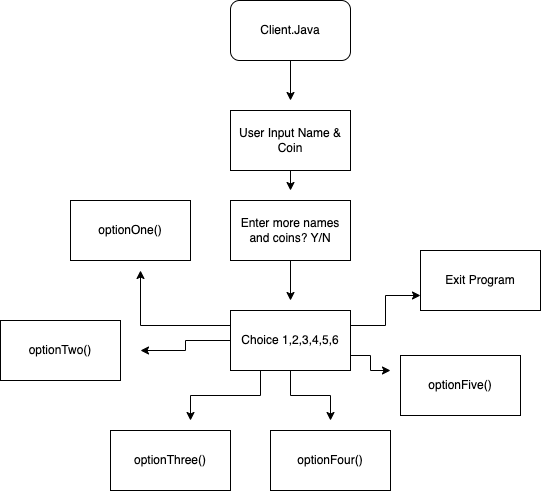


Follow prompts and enter name and coin value of the person, up to 10 records.

Step 5 -

Enter one of the options shown in the program, enter 6 to exit.

## **Structure/Design/Algorithm**

Changecoin.java UML Client.java structure

**Changecoin.java**

* Changecoin class
* name: String
* twoDollars: int
* oneDollar: int
* fiftyCents: int
* twentyCents: int
* tenCents: int
* fiveCents: int
* StudentInfo() // to display student’s info
* getName() // accessor method to get the name variable of the object
* getCoin() // accessor method to get the coin variable of the object
* calculateChange() // calculate the number of coins to be given for each denomination
* resetChange() // to reset the stored number of coins for each denomination
* checkDivisible() // check if number divisible by 5, round to nearest multiple of 5
* menu() // print menu and return choice as int
* displayChange() // display the change according to the denominations
* optionOne() // display change of a specific person
* optionTwo() // display change of person with the largest amount of coins
* optionThree() // display change of person with the smallest amount of coins
* optionFour() // calculate and display total number of coins
* optionFive() // calculate and display total coins in amount

**Pseudocode Changecoin.java Methods**

calculateChange()

1. Initialize int temp = 0
2. Repeat until coin is below 5
3. If coin is more than or equals 200, add 200 to temp and minus 200 from coin, add 1 to twoDollars variable.
4. Repeat for oneDollar, fiftyCents, twentyCents, tenCents and fiveCents, adding and deducting 100, 50, 20, 10, 5 respectively to temp and from coin. Add 1 to their respective variables.
5. Set coin equals temp.

resetChange()

1. Reset twoDollars, oneDollar, fiftyCents, twentyCents, tenCents and fiveCents back to 0.

checkDivisible()

1. Initialize int temp = coin % 10 to get the last digit of the number.
2. If last digit of the number is more than 0 and less than 3, minus the last digit off the number.
3. If last digit of the number is more than 2 and less than 5, add 5 minus the last digit of the number.
4. If last digit of the number is more than 5 and less than 8, minus last digit of the number minus 5.
5. If last digit of the number is more than 7 and less than 10, add 10 minus last digit of the number.
6. The number should be rounded off to the nearest multiple of 5.

userInput(Changecoin[] arr)

1. Define scanner, string userchoice = “Y”, string username and int amount.
2. Fill arr with empty objects.
3. Loop through the arr and ask user for name input.
4. Loop through the arr again and check if the name exist in the system.
5. Ask user for coin value.
6. Create object and add it to array with the name input and coin value.
7. Ask if user wants to add more users.
8. If input = Y, loop again, if input = N, break loop.
9. Let user know the array is full once 10 set of objects are created.

menu()

1. Define scanner and int choice.
2. Print out menu options
3. Get user choice and return as int

displayChange()

1. Initialize string msg
2. Check if each of the denomination variable is more than 0.
3. If more than 0, add amount to string msg
4. Print msg

isExisting(Changecoin[] arr, String name)

1. Initialize boolean result = false.
2. Loop through array to see if the name provided in the parameter exists in the array
3. If name exists, set result to true
4. Return result

optionOne(Changecoin[] arr)

1. Define scanner and initialize boolean exist = true
2. Ask user for name input
3. Loop through arr to see if name exists
4. If name exists, print not found
5. If name doesn’t exist, call calculateChange(), displayChange() and resetChange() methods.

optionTwo(Changecoin[] arr)

1. Initialize largestIndex=0 and largestNum=arr[0].coin
2. Loop through arr to see if coin is more than the previous object, set the index of the object with the most amount of coins as largestIndex
3. Call calculateChange(), displayChange() and resetChange() methods on arr[largestIndex].

optionThree(Changecoin[] arr)

1. Initialize smallestIndex=0 and smallestNum=arr[0].coin
2. Loop through arr to see if coin is less than the previous object, set the index of the object with the least amount of coins as smallestIndex
3. Call calculateChange(), displayChange() and resetChange() methods on arr[smallestIndex].

optionFour(Changecoin[] arr)

1. Initialize variables total, two, one, fifty, twenty, ten, five, temp = 0 and msg = “Total number of coins for each denomination”
2. Loop through array and add coins of each object to total.
3. Use while loop to calculate change for each of the denomination.
4. Add to msg based on amount of each denomination.
5. Return msg

optionFive(Changecoin[] arr)

1. Initialize variables total, two, one, fifty, twenty, ten, five, temp = 0 and msg = “Total amount for each denomination”
2. Loop through array and add coins of each object to total.
3. Use while loop to calculate change for each of the denomination.
4. Convert each of the denomination to their various amounts in cents. (i.e. 2 dollars = 200 cents)
5. Add the amount in cents to msg for each of the denomination.
6. Return msg

**Pseudocode Client.java**

1. Define as main method
2. Initialize personArr[] as array of size 10 with Changecoin class objects.
3. Create new Changecoin object “coinobj”.
4. Initialize int choice = 0
5. Call coinobj.studentInfo()
6. Print a message stating only 10 records can be entered
7. Call coinobj.userInput() with personArr as the parameter
8. Loop until choice is equal to 6.
9. Call coinobj.menu()
10. If 1 is returned, call coinobj.optionOne()
11. If 2 is returned, call coinobj.optionTwo()
12. If 3 is returned, call coinobj.optionThree()
13. If 4 is returned, call coinobj.optionFour()
14. If 5 is returned, call coinobj.optionFive()
15. If 6 is returned, print farewell message and exit the program

testing(Changecoin[] arr)

1. Add 10 Changecoin objects to the arr

## **O-O Design Concepts & Software Design Patterns**

1, Single Responsibility Rule

Most methods in the Changecoin class have only a single responsibility.

For example,

displayChange() method is solely used to check how many of each denomination is in the object and to display it as output.

menu() method is used only to display the menu as an output and to return the choice chosen. checkDivisible() method is used to check if the amount of coins is divisible by 5 and to set the value to the nearest multiple of 5.

resetChange() method is used to set all denominations to 0.

calculateChange() method is used to calculate the amount of each denomination for the particular coin value.

2. Information hiding

All the variables in the Changecoin class is set to private. This make sure that the information is hidden and only the class members can access the variables.

public class Changecoin {

private String name;

private int coin;

private int twoDollars;

private int oneDollar;

private int fiftyCents;

private int twentyCents;

private int tenCents;

private int fiveCents;

3. Encapsulation

Because the variables is set to private and only Changecoin class members can access the variables, we create a new Changecoin object in the Client class and use methods from the Changecoin class to access methods and variables from the Changecoin class.

Changecoin coinobj = new Changecoin("x",1); // creates a new Changecoin class object

coinobj.userInput(personArr); // calls methods from the Changecoin class

4. Constructors

The Changecoin class object has 2 constructors.

The default constructor with no parameters,

public class Changecoin {

private String name;

private int coin;

private int twoDollars;

private int oneDollar;

private int fiftyCents;

private int twentyCents;

private int tenCents;

private int fiveCents;

And, the parametized constructor with 2 parameters, String name and int value,

public Changecoin(String name, int value){

this.name = name;

this.coin = value;

}

5. Accessor and Mutators

The Changecoin class has various accessor and mutator methods to access and mutate the variables from outside the class.

public String getName(){

return this.name;

}

public int getCoin(){

return this.coin;

}

The getName() and getCoin() methods are used to access the name and coin variable of the class object.

public void calculateChange(){

int temp = 0;

while (this.coin >= 200){

this.coin -= 200;

temp += 200;

this.twoDollars += 1;

}

while (this.coin >= 100){

this.coin -= 100;

temp += 100;

this.oneDollar += 1;

}

while (this.coin >= 50){

this.coin -= 50;

temp += 50;

this.fiftyCents += 1;

}

while (this.coin >= 20){

this.coin -= 20;

temp += 20;

this.twentyCents += 1;

}

while (this.coin >= 10){

this.coin -= 10;

temp += 10;

this.tenCents += 1;

}

while (this.coin >= 5){

this.coin -= 5;

temp += 5;

this.fiveCents += 1;

}

this.coin = temp;

}

calculateChange() method is used to mutate the coin variable and also the variable for each of the denominators.

public void resetChange(){

this.twoDollars = 0;

this.oneDollar = 0;

this.fiftyCents = 0;

this.twentyCents = 0;

this.tenCents = 0;

this.fiveCents = 0;

}

resetChange() method mutates the denomination variables.

public void checkDivisible(){

int temp = this.coin % 10;

if (temp > 0 && temp < 3){

this.coin -= temp;

}

if (temp > 2 && temp < 5 ){

this.coin += (5-temp);

}

if (temp > 5 && temp < 8){

this.coin -= (temp-5);

}

if (temp > 7 && temp < 10){

this.coin += (10-temp);

}

}

checkDivisible() method mutates the coin variable to set coin value to nearest multiple of 5.

## **Limitations**

1. Because the program required us to store objects specifically in an Array and not an ArrayList, we could not change the size of the Array thus, there will be a maximum number of entries allowed.

## 

## 

## 

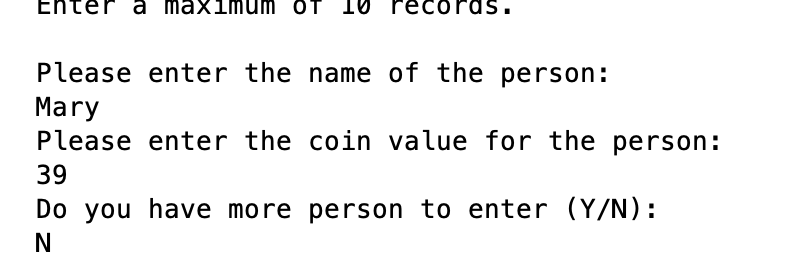
## 

## **Testing**

| **Test ID** | **Test description/justification – what is the test for and why this particular test.** | **Actual data for this test** | **Expected output** | **Actual desk check result when desk check is carried out** | **Desk check outcome – Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 1 | Test if checkDivisible() correctly round off value to nearest multiple of 5. | Coin Value 39 | 40 | 40 | Pass |
| 2 | Test if program successfully check if a name already exists in the array. | John  John | Name already exist. | Name already exist. | Pass |
| 3 | Test if change is correctly calculated for each denomination. | Coin value 50 | Fifty cents - 1 | Fifty cents - 1 | Pass |
| 4 | Test if optionTwo() gets person with largest value of coins. | John - 50  Mary - 100 | Get Mary and calculate change | Mary 100 coins  1 Dollar - 1 | Pass |
| 5 | Test if optionThree() gets person with smallest value of coins. | John - 50  Mary - 100 | Get John and calculate change | John 50 coins  50 cents - 1 | Pass |
| 6 | Test if optionFour() correctly calculates the total number of coins for each denomination | John - 50  Mary - 100 | Total = 150  1 dollar - 1  50 cents -  1 | 1 dollar - 1  50 cents - 1 | Pass |
| 7 | Test if optionFive() correctly calculates the total amount for the sum of all denominations. | John - 50  Mary - 100 | 1 dollar = 100 cents  50 cents = 50 cents | 1 dollar = 100 cents  50 cents = 50 cents | Pass |
| 8 | Test if farewell message is printed upon choosing option 6. | 6 | Farewell message is printed and program exits. | Thank you for using, good bye!  BUILD SUCCESSFUL (total time: 8 minutes 22 seconds) | Pass |

**Test ID 1**

Test if checkDivisible() correctly round off value to nearest multiple of 5.

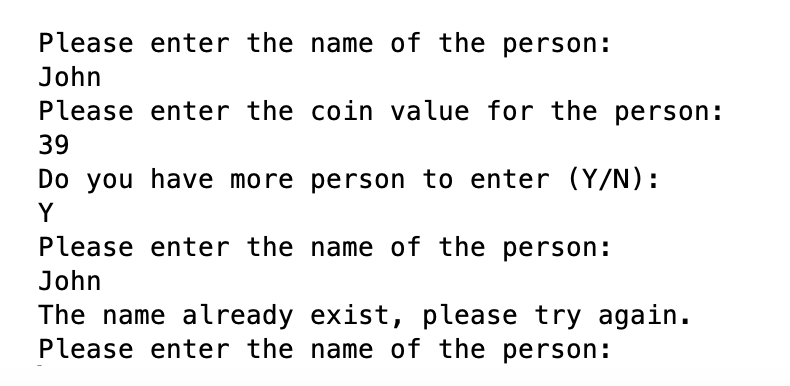


Changeclass Object with name Mary and coin value 39 is created. checkDivisible() method should return 40 to the coin value.



**Test ID 2**

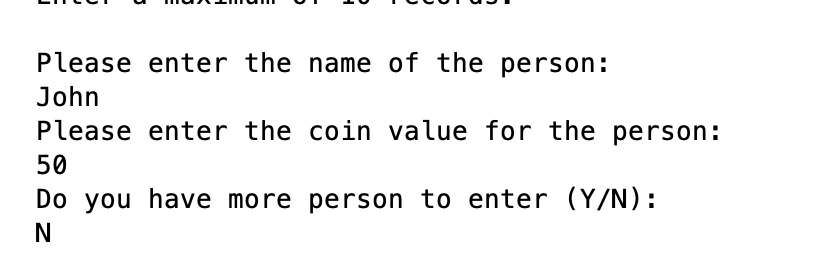
Test if program successfully check if a name already exists in the array.



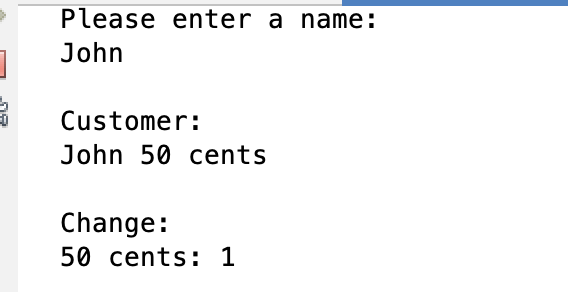
As John is already entered for entry number 1, when user attempted to enter John again, program shows the name already exists.

**Test ID 3**

Test if change is correctly calculated for each denomination.



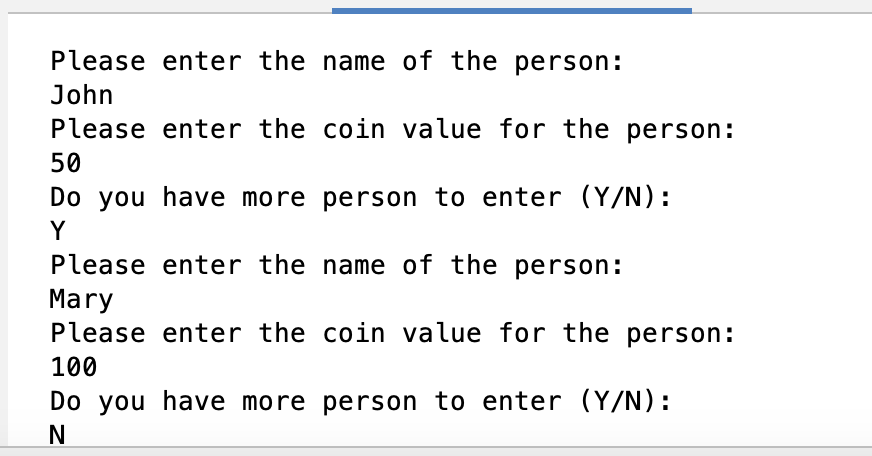
Changeclass object with 50 coins created.



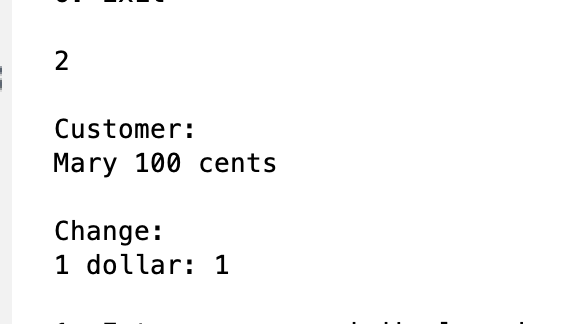
Change successfully calculated.

**Test ID 4**

Test if optionTwo() gets person with largest value of coins.



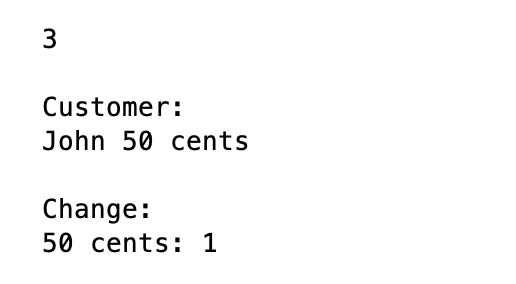
Successfully store John with 50 coins and Mary with 100 coins to the array.



Successfully defined Mary as the person with the largest amount of coins.

**Test ID 5**

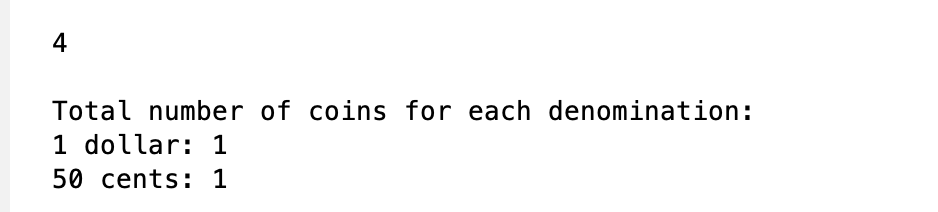
Continued from Test ID 4, check optionThree().



Successfully defined John as the person with the smallest amount of coins.

**Test ID 6**

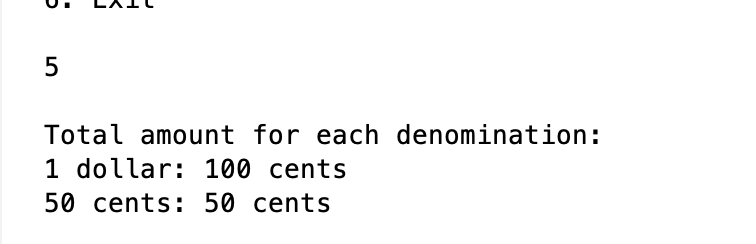
Continued from Test ID 5, test if optionFour() correctly calculates the total number of coins for each denomination.



Successfully calculated the 50 coins from John and 100 coins from Mary to get a total of 150 coins and successfully displayed the change for the 150 total coins.

**Test ID 7**

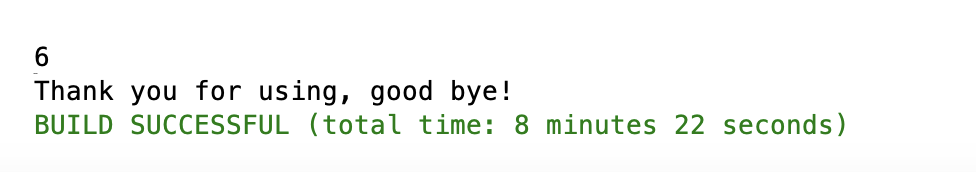
Continued from Test ID 6, test if optionFive() correctly calculates the total amount for the sum of all denominations.



Successfully calculated 150 total coins as their amount in cents.

**Test ID 8**

Check if farewell message and program exits upon entering choice 6.



Upon entering option 6, the program prints out a farewell message and exits the program.

## **Source Program Listings**

**Changecoin.java**

package assignment1;

import java.util.Scanner;

/\*\*

\*

\* @author Ang Jin Wei

\* @date 8 Feb 2023

\* @filename Changecoin.java

\* @

\*/

public class Changecoin {

private String name;

private int coin;

private int twoDollars;

private int oneDollar;

private int fiftyCents;

private int twentyCents;

private int tenCents;

private int fiveCents;

//default constructor

public Changecoin(){

this.name = "empty";

this.coin = 0;

}

//constructor with two parameters

public Changecoin(String name, int value){

this.name = name;

this.coin = value;

}

public void StudentInfo(){

String msg = "";

msg += "Name: Ang Jin Wei";

msg += "\nStudent Number: 34792195";

msg += "\nMode of Enrolment: Part-Time";

msg += "\nTutor Name: Steven Loke";

msg += "\nClass: ICT284 B";

System.out.println(msg+"\n");

}

//start of accessor methods

public String getName(){

return this.name;

}

public int getCoin(){

return this.coin;

}

//end of accessor methods

//calculate amount of coins for each denominator

public void calculateChange(){

int temp = 0;

while (this.coin >= 200){

this.coin -= 200;

temp += 200;

this.twoDollars += 1;

}

while (this.coin >= 100){

this.coin -= 100;

temp += 100;

this.oneDollar += 1;

}

while (this.coin >= 50){

this.coin -= 50;

temp += 50;

this.fiftyCents += 1;

}

while (this.coin >= 20){

this.coin -= 20;

temp += 20;

this.twentyCents += 1;

}

while (this.coin >= 10){

this.coin -= 10;

temp += 10;

this.tenCents += 1;

}

while (this.coin >= 5){

this.coin -= 5;

temp += 5;

this.fiveCents += 1;

}

this.coin = temp; //revert coin amount to original after calculation

}

//reset denominations count to 0

public void resetChange(){

this.twoDollars = 0;

this.oneDollar = 0;

this.fiftyCents = 0;

this.twentyCents = 0;

this.tenCents = 0;

this.fiveCents = 0;

}

//check divisible by 5, if not round off to nearest multiple of 5

public void checkDivisible(){

int temp = this.coin % 10; //find last digit of number

if (temp > 0 && temp < 3){

this.coin -= temp;

}

if (temp > 2 && temp < 5 ){

this.coin += (5-temp);

}

if (temp > 5 && temp < 8){

this.coin -= (temp-5);

}

if (temp > 7 && temp < 10){

this.coin += (10-temp);

}

}

//method to allow user to input names and value of coins

public Changecoin[] userInput(Changecoin[] arr){

Scanner input = new Scanner(System.in);

String userchoice = "Y";

String username;

int amount;

for (int x = 0; x<arr.length ; x++){

arr[x] = new Changecoin(); //fill array with "empty" objects

}

for (int i = 0; i<arr.length; i++){

System.out.println("Please enter the name of the person:");

username = input.nextLine();

if (isExisting(arr,username) == true){

System.out.println("The name already exist, please try again.");

i -= 1;

}

if (isExisting(arr, username) == false){

System.out.println("Please enter the coin value for the person:");

amount = input.nextInt();

arr[i] = new Changecoin(username,amount);

arr[i].checkDivisible();

if(i < 9){ //check if max array size reached

System.out.println("Do you have more person to enter (Y/N):");

input.nextLine();

userchoice = input.nextLine();

}

else{

System.out.println("\nYou have reached the maximum number of records.\n");

}

}

if (userchoice.equalsIgnoreCase("N")){

break;

}

}

return arr;

}

//menu returning choice of user

public int menu(){

Scanner input = new Scanner(System.in);

int choice;

System.out.println("\n1. Enter a name and display change to be given for each denomination");

System.out.println("2. Find the name with the largest amount and display change to be given for each denomination");

System.out.println("3. Find the name with the smallest amount and display change to be given for each denomination");

System.out.println("4. Calculate and display the total number of coins for each denomination");

System.out.println("5. Calculate and display the total amount (i.e. NOT the total number of coins) for the sum of all denominations");

System.out.println("6. Exit\n");

choice = input.nextInt();

return choice;

}

//display change based on amount of each denominator, not displaying if amount equals 0

public void displayChange(){

String msg = "\nCustomer: \n" + this.name + " " + this.coin + " cents\n\nChange:";

if (this.twoDollars != 0){

msg += "\n2 dollars: " + this.twoDollars;

}

if (this.oneDollar != 0){

msg += "\n1 dollar: " + this.oneDollar;

}

if (this.fiftyCents != 0){

msg += "\n50 cents: " + this.fiftyCents;

}

if (this.twentyCents != 0){

msg += "\n20 cents: " + this.twentyCents;

}

if (this.tenCents != 0){

msg += "\n10 cents: " + this.tenCents;

}

if (this.fiveCents != 0){

msg += "\n5 cents: " + this.fiveCents;

}

System.out.println(msg);

}

//method to check if name exists

public boolean isExisting(Changecoin[] arr, String name){

boolean result = false;

for (int i = 0; i< arr.length; i++){

if (arr[i].name.equalsIgnoreCase(name)){

result = true;

}

}

return result;

}

//enter a name and display change to be given for each denomination

public void optionOne(Changecoin[] arr){

Scanner input = new Scanner(System.in);

boolean exist = true;

System.out.println("\nPlease enter a name:");

String userinput = input.nextLine();

for (int i = 0; i<arr.length; i++){

if (arr[i].name.equalsIgnoreCase(userinput)){

arr[i].calculateChange();

arr[i].displayChange();

arr[i].resetChange();

exist = true;

break;

}

else{

exist = false;

}

}

if (exist == false){

System.out.println("\nName: " + userinput + "\nNot found\n");

}

}

//display change of the person with largest amount of coins

public void optionTwo(Changecoin[] arr){

int largestIndex=0;

int largestNum=arr[0].coin;

for (int i = 0; i < arr.length; i++){

if (arr[i].coin > largestNum){

largestNum = arr[i].coin;

largestIndex = i;

}

}

arr[largestIndex].calculateChange();

arr[largestIndex].displayChange();

arr[largestIndex].resetChange();

}

//display change of the person with smallest amount of coins

public void optionThree(Changecoin[] arr){

int smallestIndex=0;

int smallestNum=arr[0].coin;

for (int i = 0; i < arr.length; i++){

if (arr[i].coin < smallestNum && arr[i].name.equals("empty") == false){ //ignore default entry

smallestNum = arr[i].coin;

smallestIndex = i;

}

}

arr[smallestIndex].calculateChange();

arr[smallestIndex].displayChange();

arr[smallestIndex].resetChange();

}

//calculate and display total number of coins for each denomination

public void optionFour(Changecoin[] arr){

int total = 0;

String msg = "\nTotal number of coins for each denomination:";

int two = 0;

int one = 0;

int fifty = 0;

int twenty = 0;

int ten = 0;

int five = 0;

int temp = 0;

for (int i = 0; i < arr.length; i++){

total += arr[i].coin;

temp = total;

}

while (total >= 200){

total -= 200;

two += 1;

}

while (total >= 100){

total -= 100;

one += 1;

}

while (total >= 50){

total -= 50;

fifty += 1;

}

while (total >= 20){

total -= 20;

twenty += 1;

}

while (total >= 10){

total -= 10;

ten += 1;

}

while (total >= 5){

total -= 5;

five += 1;

}

if (two != 0){

msg += "\n2 dollars: " + two;

}

if (one != 0){

msg += "\n1 dollar: " + one;

}

if (fifty != 0){

msg += "\n50 cents: " + fifty;

}

if (twenty!= 0){

msg += "\n20 cents: " + twenty;

}

if (ten != 0){

msg += "\n10 cents: " + ten;

}

if (five != 0){

msg += "\n5 cents: " + five;

}

System.out.println(msg);

}

public void optionFive(Changecoin[] arr){

int total = 0;

String msg = "\nTotal amount for each denomination:";

int two = 0;

int one = 0;

int fifty = 0;

int twenty = 0;

int ten = 0;

int five = 0;

int temp = 0;

for (int i = 0; i < arr.length; i++){

total += arr[i].coin;

temp = total;

}

while (total >= 200){

total -= 200;

two += 1;

}

while (total >= 100){

total -= 100;

one += 1;

}

while (total >= 50){

total -= 50;

fifty += 1;

}

while (total >= 20){

total -= 20;

twenty += 1;

}

while (total >= 10){

total -= 10;

ten += 1;

}

while (total >= 5){

total -= 5;

five += 1;

}

two \*= 200;

one \*= 100;

fifty \*= 50;

twenty \*= 20;

ten \*= 10;

five \*= 5;

if (two != 0){

msg += "\n2 dollars: " + two + " cents";

}

if (one != 0){

msg += "\n1 dollar: " + one + " cents";

}

if (fifty != 0){

msg += "\n50 cents: " + fifty + " cents";

}

if (twenty!= 0){

msg += "\n20 cents: " + twenty + " cents";

}

if (ten != 0){

msg += "\n10 cents: " + ten + " cents";

}

if (five != 0){

msg += "\n5 cents: " + five + " cents";

}

System.out.println(msg);

}

} //end of Changecoin class

**Client.java**

package assignment1;

/\*\*

\*

\* @author Ang Jin Wei

\* @date 8 Feb 2023

\* @filename Client.java

\* @

\*/

public class Client {

public static void main(String[] args) {

Changecoin personArr[] = new Changecoin[10]; //initialize array of size 10

Changecoin coinobj = new Changecoin(); //create new Changecoin class object

int choice = 0;

//testing(personArr);

coinobj.StudentInfo();

System.out.println("Enter a maximum of 10 records.\n");

coinobj.userInput(personArr);

while (choice != 6){

choice = coinobj.menu();

switch(choice){

case 1:

coinobj.optionOne(personArr);

break;

case 2:

coinobj.optionTwo(personArr);

break;

case 3:

coinobj.optionThree(personArr);

break;

case 4:

coinobj.optionFour(personArr);

break;

case 5:

coinobj.optionFive(personArr);

break;

}

if (choice < 1 || choice > 6){

System.out.println("\nInvalid choice, please try again.\n");

}

}

System.out.println("Thank you for using, good bye!");

}

//hardcoded method to store 10 Changecoin objects in the array

public static void testing(Changecoin[] arr){

arr[0] = new Changecoin("John",95);

arr[1] = new Changecoin("Mary", 65);

arr[2] = new Changecoin("Peter", 45);

arr[3] = new Changecoin("Fred", 85);

arr[4] = new Changecoin("Tom", 90);

arr[5] = new Changecoin("Jane", 40);

arr[6] = new Changecoin("May", 75);

arr[7] = new Changecoin("Alfred", 30);

arr[8] = new Changecoin("Carmen", 15);

arr[9]= new Changecoin("Poka", 25);

}

}