**Make Change Program (Part A)**

**Imports:**

java.util.Scanner

**Global Variables:**

Scanner sc – scanner object to read System.in

**Methods:**

static void main(String[] args) – process control with loop for multiple runs

static int getCents() – obtain and validate input of change being requested

static void makeChange(int cents) – calculate and display coins given in change

**Process:**

main(String[] args):

local variables: int cents – variable for change amount being requested

1. Display welcome message
2. getCents 🡪 cents
3. while loop on cents != 0
   1. makeChange(cents)
   2. getCents 🡪 cents
4. Display end message

getCents():

local variables: int c – cents input value to return

1. do - loop to control input/validation process
   1. Try
      1. How much change? 🡪 c
      2. c < 0? 🡪 error message
      3. c > 100? 🡪 error message
   2. Catch
      1. Illegal input message
      2. Clear input buffer
      3. Set c to sentinel value (to force loop to repeat)

while (c < 0 or c > 100)

1. return c;

makeChange(int cents):

local variables: int q,d,n,p – for quarters, dimes, nickels, pennies to give

int r – working variable for ‘remaining cents’ to give

1. set r = cents
2. Determine Quarters to give:
   1. q = r / 25 (integer division)
   2. adjust r to new remaining value: r = r – (q \* 25)
3. Repeat step 2 for dimes, nickels (in that order)
4. Remaining r after step 3 is given as pennies
5. Print results

**MakeChange Program Changes for: Part B**

**Global Variables:**

Add: int qoh, doh, noh, and poh for Quarters-on-Hand, Dimes-on-Hand, Nickels-on-Hand, and Pennies-on-Hand

**Methods:**

Add: int getCoins(String cointype) – to validate and return coins on hand for each ‘cointype’

**Specific method changes:**

**main():**

Add logic to request each coin type using getCoins() and place in proper global variable (qoh, doh, etc.)

**Int getCoins(String cointype):**

Code is virtually the same as getCoins method from prior assignment (calcChange)

**makeChange(int cents):**

Changes to algorithm: basic process is the same, except after determining each coin count, value must be checked against coins on hand, and adjusted accordingly. So, for prior step 2 you would now have:

Determine Quarters to give:

* 1. q = r /25 (integer division)
  2. if q > qoh, adjust q to equal qoh
  3. adjust r as: r = r – (q \* 25)

The final step (print results) must also check that r has been worked down to zero: if not, perfect change could not be given from the available set of coins.