Reflection log

```
1  package Mastery;
2
3  import java.util.Scanner;
6
7  public class Mysavings {
8
```

The code defines the Mysavings class within the Mastery package, indicating its purpose relates to savings. It also imports the Scanner class, preparing the program to handle user input from the console.

```
9          @SuppressWarnings("resource")
0          public static <piggyBank> void main(String[] args) {
1              //links piggybank to mysavings
2              piggybank pb = new piggybank();
```

The code defines the main method with a SuppressWarnings annotation to ignore resource-related warnings. It creates a piggybank object, linking it to the Mysavings class for further operations.

```
//links piggybank to mysavings
piggybank pb = new piggybank();

// Lets the user input values and makes it so that decimals only go to the hundredths
Scanner in = new Scanner(System.in);
DecimalFormat deca = new DecimalFormat("#.##");
```

This portion of the code initializes two main objects to support user interaction and data formatting. The piggybank pb = new piggybank(); line creates a new instance of the piggybank class, presumably linking it to manage savings or similar functionality. Additionally, the Scanner object enables user input via the console, while the DecimalFormat ensures that numerical values are formatted to two decimal places, enhancing readability and consistency in financial calculations.

```
int choice = 0;

do {
//prompt user to enter a number to pick which type of metric conversion they want
    System.out.println("1. Show total in bank");
    System.out.println("2. Add a penny");
    System.out.println("3. Add a nickel");
    System.out.println("4. Add a dime");
    System.out.println("5. Add a quarter");
    System.out.println("6. Take money out of your bank");
    System.out.println("Enter 0 to quit");
    System.out.print("Enter your choice: ");
    choice = in.nextInt();
```

This part of the code represents a menu-driven program that provides users with various options for interacting with a virtual "bank." The do-while loop ensures that the menu is

displayed repeatedly until the user decides to quit by entering 0. Each menu item corresponds to a specific action (like adding different denominations or showing the total), which the program processes based on the user's numeric input

```
//switch case for each metric conversion choice
switch (choice) {

//Show total in bank when user picks 1
```

This segment uses a switch statement to handle user input from the menu. Based on the value of choice, the program will execute the corresponding block of code to perform a specific operation, such as showing the total or adding coins.

```
39                  case 1:
40
41                      //takes user input and applies inchestocm method to it
42                      System.out.println("Total in bank: $" + deca.format(pb.bankTotal()));
43                      break;
44
45                    //Add a penny when user picks 2
46                  case 2:
47                    //takes user input and applies cmtoinches method to it
48                      pb.penny(1);
49                      System.out.println("Added 1 penny");
50                      break;
51
52                    //Add a nickel when user picks 3
53                  case 3:
54                    //takes user input and applies feettocm method to it
55                      pb.nickel(1);
56                      System.out.println("Added 1 nickel");
57                      break;
58
59                    //Add a dime when user picks 4
60                  case 4:
61
62                    //takes user input and applies cmtofeet method to it
63                      pb.dime(1);
64                      System.out.println("Added 1 dime");
65                      break;
66
67                    //Add a quarter when user picks 5
68                  case 5:
69                    //takes user input and applies yardstom method to it
70                      pb.quarter(1);
71                      System.out.println("Added 1 quarter");
72                      break;
73
74                    //Take money out of piggy bank when user picks 6
75                  case 6:
76                    //takes user input and applies mtoyards  method to it
77                      System.out.println("You took $" + deca.format(pb.bankTotal()) + " out of the bank" );
78                      pb.takeOut();
79                      break;
80
81                    //runs case 7 when user picks 7
82                  case 0:
83                    //takes user input and applies miletokm method to it
84                      System.out.println("Quit successful");
85                      break;
86              }
87              }
88              while (choice != 0);
89
90              }
91          }
```

This code implements the logic for the user's choices using a switch statement. Each case corresponds to a menu option, such as adding coins to the bank or showing the total amount, and executes specific methods from the pb object. For example, the cases for adding a penny, nickel, dime, or quarter call respective methods (penny(), nickel(), etc.) and print a confirmation message. The program also handles withdrawing all money (case 6) and exiting the program (case 0) with appropriate messages. The do-while loop ensures the menu is continuously displayed until the user opts to quit by entering 0.