Credit Name: CSE 2140 2nd Language Programming
Assignment Name: Chapter 7 Portfolio

### Error Log Entry

What error message did you encounter (if any)?
This code was fairly simple and straightforward to understand. I, however, towads the end got a little stuck on how to format and properly perform some tasks.
What unexpected behavior did your program exhibit?
When I ran my program, my program would start off showing the different options and prompting the user on their choice. However, when I would enter the designated number corresponding the favored choice the program would:
1. Not repeat the "Enter choice" statement in the order and way that I wanted it too
2. Even when I didn't choose the "1. Show total in bank." option, it would show the code line for this option nonetheless.

What caused the issue? (e.g., syntax error, logic error, incorrect function usage, etc.)
It was more of a logic issue, as I later relaized when looking back on my code. The error was caused because I mixed up the user of ! with ==. Because of this, the program thought that when user types anyhting BUT 1 it will display. I also realized the way I formatted and displayed the total option (regardless on the wrong timing) was also done incorrectly which I fixed.

Include a screenshot of specific lines of code.

```
int choice;

while (choice != 1) {
    System.out.println("Enter choice: .");
    choice = input.nextInt();
}
```

```
testmySavings [Java Application] C:\Program Files\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take 0 to quit.
Enter choice:
5
Your total in bank:  $%.2f\n", bank.getTotal());Enter choice:
```

How did you fix the issue?
I changed all my if statements to == instead of =! and I cleaned up the code for when it displays the total. I also in this round of checking realized that a do while loop would be better to use than the while. All these errors were then corrected.

Provide the corrected code or solution using a screenshot.

```
import java.util.Scanner;

public class testmySavings {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner input = new Scanner(System.in);
        Piggybank bank = new Piggybank(); // Now constructor works
        double totalWithdrawn = 0.0;


        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. ");
        System.out.println("Enter 0 to quit");

        int choice;

        do {
            System.out.print("\nEnter choice: ");
            choice = input.nextInt();

            if (choice == 1) {
                System.out.printf("Your total in bank: $%.2f\n", bank.getTotal());
            } else if (choice == 2) {
                bank.AddPennies(1);

            } else if (choice == 3) {
                bank.AddNickels(1);

            } else if (choice == 4) {
                bank.AddDimes(1);

            } else if (choice == 5) {
                bank.AddQuarters(1);

                }
            else if (choice == 6) {
                double withdrawn = bank.withdrawAll(); // Withdraw all money
                totalWithdrawn += withdrawn;

            } else if (choice == 0) {
                System.out.printf("\nFinal balance: $%.2f\n", bank.getTotal());
                System.out.printf("Total withdrawn: $%.2f\n", totalWithdrawn);
                break;
            }

        } while (choice != 0);

    }
```

Piggybank.java | Food.java | FoodTester.java | teststudent....

```
package Mastery;

public class Piggybank {
 private int pennies;
 private int nickels;
 private int dimes;
 private int quarters;

 public Piggybank() {
      pennies = 0;
      nickels = 0;
      dimes = 0;
      quarters = 0;
 }
 public void AddPennies(int count) {
      pennies += count;
     }
public void AddNickels(int count) {
     nickels += count;
}

public void AddDimes(int count) {
     dimes += count;
}

public void AddQuarters(int count) {
     quarters += count;
}

public double getTotal() {
     return pennies * 0.01 + nickels * 0.05 + dimes * 0.10 + quarters * 0.25;
}

// Withdraw all money at once
public double withdrawAll() {
     double total = getTotal();
     pennies = 0;
     nickels = 0;
     dimes = 0;
     quarters = 0;
     return total;
}
}
```

TEST RUN:

```
<terminated> testmySavings [Java Application] C:\P
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out.
Enter 0 to quit

Enter choice: 2

Enter choice: 2

Enter choice: 2

Enter choice: 2

Enter choice: 2

Enter choice: 5
```

```
Enter choice: 5

Enter choice: 5

Enter choice: 6

Enter choice: 1
Your total in bank: $0.00

Enter choice: 0

Final balance: $0.00
Total withdrawn: $0.55
```