**Lab 8 Pre-Lab**

**The following exercises must be completed before you come to lab. Your instructor will check your pre-lab exercises at the beginning of the lab period. Completion of the pre-lab is worth 10 points of the total 50 points for the lab.**

**Part 1**

Write the pseudocode that determines the change to be dispensed from a vending machine. An item in the machine can cost between 25 cents and a dollar, in 5-cent increments (25, 30, 35, . . . 90, 95, or 100), and the machine accepts only a single dollar bill to pay for the item. Requirements and example output. ***(bold items are user input)***

FIRST we will declare and assign 5 variables of the type Integer.

TAKE user input of the amount by using Scanner.

IF the amount is divisible by 5 and is greater or equal to 25 and less than 100 then it’s a valid amount.

ELSE invalid amount

DEDUCT the original amount from 100.

COMPUTE quarters, nickles and dimes.

IF 1 quarter , print quarter.

IF more than 1 but print quarters.

IF 1 dime then print dime.

IF more then 1 print dimes

IF 1 penny print penny.

IF more than 1 then print pennies.

* Only coins with a non-zero value should be displayed.

Item price must be 25 cents to a dollar, in 5-cent increments.

Enter item price: **75**

You bought an item for 75 cents and gave me a dollar.

Your change is:

1 quarter

* If more than one coin is returned, the word should be plural with an “s” at the end.
* If only one coin is returned, the word should not have an “s” at the end.

Item price must be 25 cents to a dollar, in 5-cent increments.

Enter item price: **35**

You bought an item for 35 cents and gave me a dollar.

Your change is:

2 quarters

1 dime

1 nickel

* Display the change only if a valid price is entered (no less than 25 cents, not more than 100 cents, and an integer multiple of 5 cents). Otherwise, display separate error messages for any of the following invalid inputs: a cost under 25 cents, a cost that is not an integer multiple of 5, and a cost that is more than a dollar.

Example 1

Item price must be 25 cents to a dollar, in 5-cent increments.

Enter item price: **5**

Cost is invalid - it must be at least 25 cents.

Example 2

Item price must be 25 cents to a dollar, in 5-cent increments.

Enter item price: **150**

Cost is invalid - it must be less than 100 cents.

Example 3

Item price must be 25 cents to a dollar, in 5-cent increments.

Enter item price: **73**

Cost is invalid - it must be evenly divisible by 5

Example 4

Item price must be 25 cents to a dollar, in 5-cent increments.

Enter item price: **153**

Cost is invalid - it must be less than 100 cents.

Cost is invalid - it must be evenly divisible by 5

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Cost** | **Change** | **Output** |
| Valid  only quarters in change | 25 | 3 | 3 quarters |
| Valid  only quarter(s) & dime(s) | 65 | 3 | 1 quarter  1 dime |
| Valid  quarter(s), dime(s), & nickel(s) | 35 | 4 | 2 quarters  1 dime  1 nickel |
| Valid  1 quarter change | 75 | 1 | 1 quarter |
| Invalid  less than 25 | 5 | 0 | Invalid input |
| Invalid  more than 100 | 125 | 0 | Invalid Input |
| Invalid  not multiple of 5 | 121 | 0 | Invalid Input |
| Invalid  100 |  |  |  |

**Part 2**

Write the pseudocode to determine geographic area given a zip code with the following requirements:

|  |  |
| --- | --- |
| **Zip code begins with** | **Geographic Area** |
| 0 – 3 | East Coast |
| 4 – 6 | Central Plains |
| 7 | South |
| 8 – 9 | West |
| other | invalid zip code |

NO, instance Variables.

ONE method with a parameter of String type.

IF, the value at index of 0 of a variable is between 0 and 3 then print East Coast.

IF, the value at index of 0 of a Variable is between 4 and 6 i.e. greater than 3 and less than or equal to 6 then print Central Plains.

IF, value at index of 0 of a variable is equal to 7 then print south.

IF, value at index of 0 of a variable is equal to 8 and equal to 9 then print west.

IF, value at index of 0 of a variable is greater than 9 then its an invalid zip code

**Debugging**

Read “Using the Eclipse Debugger” and answer the following questions:

What is a syntax error? How does Eclipse tell you there is a syntax error in your code?

**Syntax**errors result from using the language rules incorrectly which means that the typing something or using a syntax which a language does not support. Syntax errors are detected at compile time. The compiler detects errors in syntax and reports them with problem markers and in the problem view.

What is a runtime error?

The run time errors are the errors which are learnt at the run time. The java generally depends the run time by throwing at exception at the run time and displays which line of code is having a problem. The runtime errors in eclipse are displayed in the stack trace in the console view.

What is a logic error?

Logic errors are not detected at the compile as well as at the run time because these are errors which are undetected by Java because they are human errors which leads to incorrect output or can even lead to odd program behavior. For instance – Infinite Loop.

What is the difference between the two types of red X’s that can be indicated by Eclipse?

Simple red X



It is an error caused where the Eclipse will display the message why there is an error.

Lightbulb red X



In case of the light bulb, Eclipse provides you suggestion and you can select the suggestion to get rid of the error if you know the root cause of the error.

What does a breakpoint do?

The breakpoints basically help in inspecting your program and it stops the program execution once the breakpoint is encountered to view the variables inside a loop or conditions.

How do you set a breakpoint?

There are two ways to set the breakpoints

First open the debug mode and then one can toggle the breakpoints in two ways.

·         Double click on the left margin on the line number where you want to debug

·         From the menu go to Run -> Toggle Breakpoint

How do you remove a breakpoint?

You can remove the toggle point in the similar way as you set them.

·         Double click on the left margin on the line number where you want to debug

·         From the menu go to Run -> Toggle Breakpoint

One can also use the right click and go on the line number one want to add or remove the breakpoint.

What is the difference between the Step Into and Step Over debug controls?

Step Into Debug Controls: Execution of the program goes to the next line of code and if there is a method called then it goes into the method body.

Step Over Debug Controls: it is used to step over the next method call without entering it and the method is never stepped into it and is executed normally.