Result and Explanation

1. An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each gizmo weighs 112 grams. Write a program that reads the number of widgets and the number of gizmos in an order from the user. Then your program should compute and display the total weight of the order.

Solution: To compute the total weight of order we need to build some formulae/logic, Suppose if user is ordering 3 widgets and 2 gizmo the we have to calculate (3 \* 75 grams (number of widgets \* weight of one widgets)) +( 2 \* 112 grams(number of gizmo \* weight of one gizmo) ), add both for total weight.

Output:

A screenshot of a computer

Description automatically generated with medium confidence

1. Pretend that you have just opened a new savings account that earns 4 percent interest per year. The interest that you earn is paid at the end of the year and is added to the balance of the savings account. Write a program that begins by reading the amount of money deposited into the account from the user. Then your program should compute and display the amount in the savings account after 1, 2, and 3 years. Display each amount so that it is rounded to 2 decimal places.

Solution: To solve this question code should know what compound interest is? Compound interest is interest calculated on the initial principle plus all prior periods' interest on a savings account.

Here we must add 4% interest yearly to an initial amount. So, the logic will be

new\_initial\_amount= previous \_initial\_amount +previous \_initial\_amount \* 4% interest---this will follow for next years too. Every year new\_initial\_amount

and previous \_initial\_amount will update.

Output:

Text

Description automatically generated with medium confidence

1. Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads several feet from the user, followed by several inches. Once these values are read, your program should compute and display the equivalent number of centimeters. Hint: One foot is 12 inches. One inch is 2.54 centimeters.

Solution: Here we are taking a string input from user, user could enter 2.3, 2.0 it means 2 feet and 3 inches so how to calculate the total height in centimeter

First convert string value into float and separate integer and decimals.

Then calculate the number of inches 1 feet=12 inches and 1 inch = 2.54 cm.

e.g., input is 2.5 then after separating the string value into float 2.0 (feet)and 5.0 (inches), furthermore, multiply 2.0 feet \* 12 inches + 5.0(adding remaining inches) inches =29.0 inches. Now simply calculate total\_inches\* 2.54 = 73.66 cm’s.

Output:

A screenshot of a computer

Description automatically generated with medium confidence

1. Consider the software that runs on a self-checkout machine. One task that it must be able to perform is to determine how much change to provide when the shopper pays for a purchase with cash. Write a program that begins by reading several cents from the user as an integer. Then your program should compute and display the denominations of the coins that should be used to give that amount of change to the shopper. The change should be given using as few coins as possible. Assume that the machine is loaded with pennies, nickels, dimes, quarters, loonies and toonies.

Solution: User is giving int input as cents. So here if you know the concept of divisor, dividend, quotient and remainder then its very easy to solve this problem. Observe below diagram and example.

Divisor

Quotient

1

Divident

267

200

-200

Remainder

67

We are using two operators in code

1. // - Floor division (it gives quotient as output)
2. % - Modulus (it gives remainder as output)

e.g., User gave input 267 cents, Now 267 cents is greater than 1 toonies(200 cents)

1. Divide 267 with 200 So, output is 1 tonnies (267//200)(quotient) and remainder will be store as remaining\_cents=67 after implementing modulus (267%200).
2. Now 67 cents are less than 1 loonies(100 cents) so it will return 0 output and amount will be same 67 after(67%100).
3. The remaining amount is greater than 1 quarter(25 cents). It will return 2 quarter(50 cents) as output after (67//25) and the again (67%25) so we can store remaining amount 17 cents(remainder).
4. Now 17 cents are greater than 1 dimes(10 cents) . It will return 1 dimes as output after (17//10).
5. The remaining amount is 7 now and its greater than 1 nickels(5 cent).

It will return 1 nickels(5 cents) as output after (7//5) and the again (7%5) so we can store remaining amount 2 cents(remainder).

In last if the cents are less than 5 then simply display the remaing amount in pennies, e.g., 2 pennies

Output:

Graphical user interface, text

Description automatically generated