Chapter 3 homework

4. Chris Johanson wants a program that calculates and displays a 10%, 15%, and 20% tip on his total restaurant bill. First, create an IPO chart for this problem, and then desk-check the algorithm twice, using $35.80 and $56.78 as the total bill. After desk-checking the algorithm, list the input, processing, and output items in a chart similar to the one shown in Figure 3-23, and then enter the appropriate C++ decla-ration statements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| double totalBill | 1. Declare variables 2. Get input for totalBill 3. Calculate bill and 10% tip tip1Bill = totalBill \*1.10 4. Calculate bill and 15% tip tip2Bill = totalBill \* 1.15 5. Calculate bill and 20% tip tip3Bill = totalBill \* 1.20 6. Display output | tip1Bill  tip2Bill  tip3Bill |

Code-

//Dylan Nguyen, 9/8/20, Ch.3 Homework #4

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

//declare variables

double totalBill, tip1Bill, tip2Bill, tip3Bill;

//get input

cout << "Enter the total bill: $";

cin >> totalBill;

//process, setprecision(2) for 2 decimal places

tip1Bill = totalBill \* 1.1;

tip2Bill = totalBill \* 1.15;

tip3Bill = totalBill \* 1.2;

cout << fixed << setprecision(2);

cout << "Total bill with 10% tip: $" << tip1Bill << endl;

cout << "Total bill with 15% tip: $" << tip2Bill << endl;

cout << "Total bill with 20% tip: $" << tip3Bill << endl;

system("pause");

return 0;

}

5. Tile Limited wants a program that allows its salesclerks to enter the length and width (both in feet) of a rectangle and the price of a square foot of tile. The program should calculate and display the area of the rectangle and the total price of the tile. First, create an IPO chart for this problem, and then desk-check the algorithm twice. For the ﬁrst desk-check, use 10 feet, 12 feet, and $2.39 as the length, width, and tile price, respectively. For the second desk-check, use 5.5 feet, 10.5 feet, and $3.50. After desk-checking the algorithm, list the input, processing, and output items in a chart similar to the one shown in Figure 3-23, and then enter the appropriate C++ declaration statements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| length  width  tilePrice | 1. Declare variables 2. Get input 3. Calculate area of rectangle areaRect = length \* width 4. Calculate total price of tile totalPrice = tilePrice \* areaRect 5. Display output | areaRect  totalPrice |

Code-

//Dylan Nguyen, 9/8/20, Ch.3 Homework #5

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

//declare variables

double length, width, tilePrice, areaRect, totalPrice;

//get input

cout << "Enter length: ";

cin >> length;

cout << "Enter width: ";

cin >> width;

cout << "Enter tile price: $";

cin >> tilePrice;

//process

areaRect = length \* width;

totalPrice = tilePrice \* areaRect;

//output

cout << "Area: " << areaRect << endl;

cout << fixed << setprecision(2);

cout << "Total price: $" << totalPrice << endl;

system("pause");

return 0;

}

6. Builders Inc. wants a program that allows its salesclerks to enter the diameter of a circle and the price of railing material per foot. The program should calculate and display the total price of the railing material. Use 3.1416 as the value of pi. First, create an IPO chart for this problem, and then desk-check the algorithm twice. For the ﬁrst desk check, use 35 feet as the diameter and $2 as the price per foot. For the second desk-check, use 15.5 and $3.50. After desk-checking the algorithm, list the input, processing, and output items in a chart similar to the one shown in Figure 3-23, and then enter the appropriate C++ declaration statements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| diameter  railPrice | 1. Declare variables 2. Get input 3. Calculate totalRailing = PI \* diameter 4. Calculate totalPrice totalPrice = railPrice \* totalRailing 5. Display output | totalPrice |

Code-

//Dylan Nguyen, 9/8/20, Ch3 Homework #6

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

//declare

double diameter, railPrice, PI, totalPrice, totalRailing;

PI = 3.1416;

//input

cout << "Enter Diameter: ";

cin >> diameter;

cout << "Enter Rail Price: $";

cin >> railPrice;

//process

totalRailing = diameter \* PI;

totalPrice = totalRailing \* railPrice;

//output

cout << fixed << setprecision(2);

cout << "Total Rail Price: $" << totalPrice << endl;

system("pause");

return 0;

}

7. Currency Traders wants a program that converts American dollars to British pounds, Mexican pesos, and Japanese yen and then displays the results. Use the following conversion rates for one American dollar: 0.571505 British pounds, 10.7956 Mexican pesos, and 112.212 Japanese yen. First, create an IPO chart for this problem, and then desk-check the algorithm twice, using 1000 and 50 as the number of American dollars. After desk-checking the algorithm, list the input, processing, and output items in a chart similar to the one shown in Figure 3-23, and then enter the appropriate C++ declaration statements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| dollarAmount | 1. Declare variables 2. Get input 3. Calculate poundAmount poundAmount = dollarAmount \* 0.571505 4. Calculate pesoAmount pesoAmount = dollarAmount \* 10.7956 5. Calculate yenAmount dollarAmount \* 112.212 6. Display output | poundAmount  pesoAmount  yenAmount |

Code-

//Dylan Nguyen, 9/8/20, Ch3 Homework #7

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

//declare

double dollarAmount, poundAmount, pesoAmount, yenAmount;

//input

cout << "Enter US Dollar Amount: $";

cin >> dollarAmount;

//process

poundAmount = dollarAmount \* 0.571505;

pesoAmount = dollarAmount \* 10.7956;

yenAmount = dollarAmount \* 112.212;

//output

cout << fixed << setprecision(2);

cout << "US $" << dollarAmount << " in Pounds: " << poundAmount << endl;

cout << "US $" << dollarAmount << " in Pesos: " << pesoAmount << endl;

cout << "US $" << dollarAmount << " in Yen: " << yenAmount << endl;

system("pause");

return 0;

}