Students:
Section 2.9 is a part of 2 assignments: CSC108 CH02.1-2.10 C2A

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Includes: CA
Due: 02/04/2025, 11:59 PM EST

2.9 Constant variables

A good practice is to minimize the use of literal numbers in code. One reason is to improve code readability. newPrice = origPrice - 5 is less clear than newPrice = origPrice - priceDiscount. When a variable represents a literal, the variable's value should not be changed in the code. If the programmer precedes the variable declaration with the keyword const, then the compiler will report an error if a later statement tries to change that variable's value. An initialized variable whose value cannot change is called a ***constant variable***. *A common convention, or good practice, is to name constant variables using upper case letters with words separated by underscores, to make constant variables clearly visible in code.*

Figure 2.9.1: Constant variable example: Lightning distance.

```
#include <iostream>
using namespace std;

/*
 * Estimates distance of lightning based on seconds
 * between lightning and thunder
 */

int main() {
   const double SPEED_OF_SOUND    = 761.207;  // Miles/hour (sea level)
   const double SECONDS_PER_HOUR = 3600.0;   // Secs/hour
   double secondsBetween;
   double timeInHours;
   double distInMiles;

   cout << "Enter seconds between lightning and thunder: ";
   cin  >> secondsBetween;

   timeInHours = secondsBetween / SECONDS_PER_HOUR;
   distInMiles = SPEED_OF_SOUND * timeInHours;

   cout << "Lightning strike was approximately" << endl;
   cout << distInMiles << " miles away." << endl;

   return 0;
}
```

```
Enter seconds between lightning and thunder: 7
Lightning strike was approximately
1.48012 miles away.

...

Enter seconds between lightning and thunder: 1
Lightning strike was approximately
0.211446 miles away.
```

Feedback?

PARTICIPATION ACTIVITY 2.9.1: Constant variables.

Which of the following statements are valid declarations or uses of a constant integer variable named STEP_SIZE? Assume that variables totalStepHeight and numSteps have previously been declared as integers.

1) `int STEP_SIZE = 5;`
   - True
   - False

2) `const int STEP_SIZE = 14;`
   - True
   - False

3) `totalStepHeight = numSteps * STEP_SIZE;`
   - True
   - False

4) `STEP_SIZE = STEP_SIZE + 1;`
   - True
   - False

Feedback?

CHALLENGE ACTIVITY 2.9.1: Using constants in expressions. Full screen

620890.5010016.qx3zqy7

The cost to ship a package is a flat fee of 75 cents plus 25 cents per pound. Organize the correct code statements to:

1. Declare a constant integer variable named CENTS_PER_POUND and initialize the constant with the value 25.
2. Read the shipping weight from input and store the weight into shipWeightPounds.
3. Using the constants FLAT_FEE_CENTS and CENTS_PER_POUND, assign shipCostCents with the cost of shipping a package weighing shipWeightPounds.

Click here for example

Note: Not all code statements on the left will be used in the final solution.

How to use this tool

Unused

```
const int CENTS_PER_POUND = 25;
constant int CENTS_PER_POUND = 25;
int centsPerPound = 25;
shipCostCents = FLAT_FEE_CENTS + (CENTS_PER_POUND * shipWeight
shipCostCents = flatFeeCents + (centsPerPound * shipWeightPour
cin >> shipWeightPounds;
```

main.cpp Load default template...

```
#include <iostream>
using namespace std;

int main() {
   int shipWeightPounds;
   int shipCostCents = 0;
   const int FLAT_FEE_CENTS = 75;

   cout << "Weight(lb): " << shipWeightPounds;
   cout << ", Flat fee(cents): " << FLAT_FEE_CENTS;
   cout << ", Cents per lb: " << CENTS_PER_POUND << endl;
   cout << "Shipping cost(cents): " << shipCostCents << endl;

   return 0;
}
```

Check

Feedback?

CHALLENGE ACTIVITY 2.9.2: Constant variables.

620890.5010016.qx3zqy7

Start

Complete the declaration of the constant integer variable WEEKS_PER_YEAR.

The program then reads integer numYears from input and converts the number of years to number of weeks, using WEEKS_PER_YEAR.

Ex: If the input is `11`, then the output is:

```
11 years = 572 weeks
```

```
#include "testcode.h"  // For code testing purposes
#include <iostream>
using namespace std;

int main() {
   /* Your code goes here */ WEEKS_PER_YEAR = 52;
   int numYears;
   int numWeeks;

   cin >> numYears;

   numWeeks = numYears * WEEKS_PER_YEAR;

   cout << numYears << " years = " << numWeeks << " weeks" << endl;

   Runtests();  // Testing code
```

1 2

Check Next level

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH02.1-2.10 C2A

41 / 41 points

Due: 02/04/2025, 11:59 PM EST

41 / 41 points submitted to BlackboardLearn

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Completion details