Students:
Section 5.12 is a part of 2 assignments: CSC108 CH05.8-5.16 C5B
Includes: CA
Due: 04/15/2025, 11:59 PM EDT
Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. Learn more

5.12 Default parameter values

Sometimes a function's last parameter (or last few) should be optional. A function call could then omit the last argument, and instead the program would use a default value for that parameter. A function can have a ***default parameter value*** for the last parameter(s), meaning a call can optionally omit a corresponding argument.

Figure 5.12.1: Parameter with a default value.

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

// Function prints date in two styles (0: American (default), 1: European)
void PrintDate(int currDay, int currMonth, int currYear, int printStyle = 0) {

   if (printStyle == 0) {      // American
      cout << currMonth << "/" << currDay << "/" << currYear;
   }
   else if (printStyle == 1) { // European
      cout << currDay << "/" << currMonth << "/" << currYear;
   }
   else {
      cout << "(invalid style)";
   }
}

int main() {

   // Print dates given various style settings
   PrintDate(30, 7, 2012, 0);
   cout << endl;

   PrintDate(30, 7, 2012, 1);
   cout << endl;

   PrintDate(30, 7, 2012); // Uses default value for printStyle
   cout << endl;

   return 0;
}
```

```
7/30/2012
30/7/2012
7/30/2012
```

Feedback?

The fourth (and last) parameter has a default value: `int printStyle = 0`. If a function call does not provide a fourth argument, then the style parameter is 0.

The same can be done for other parameters, as in: `void PrintDate(int currDay = 1, int currMonth = 1, int currYear = 2000, int printStyle = 0)`. Because arguments are matched with parameters based on their ordering in the function call, only the last arguments can be omitted. The following are valid calls to this PrintDate() function having default values for all parameters:

Figure 5.12.2: Valid function calls with default parameter values.

```
PrintDate(30, 7, 2012, 0); // No defaults
PrintDate(30, 7, 2012);    // Defaults:                    style=0
PrintDate(30, 7);          // Defaults:              year=2000, style=0
PrintDate(30);             // Defaults:       month=1, year=2000, style=0 (strange, but valid)
PrintDate();               // Defaults: day=1, month=1, year=2000, style=0
```

Feedback?

If a parameter does not have a default value, then failing to provide an argument generates a compiler error. Ex: Given: void PrintDate(int currDay, int currMonth, int currYear, int printStyle = 0). Then the call PrintDate(30, 7) generates the following error message from g++.

Figure 5.12.3: Compiler error if parameters corresponding to omitted arguments don't have default values.

```
fct_defparm.cpp: In function int main():
fct_defparm.cpp:5: error: too few arguments to function void PrintDate(int, int, int)
fct_defparm.cpp:22: error: at this point in file
```

Feedback?

PARTICIPATION ACTIVITY 5.12.1: Function parameter defaults.

Given:

```
void CalcStat(int num1, int num2, int num3 = 0, char usrMethod = 'a') { ... }
```

1) A compiler error will occur because only an int parameter can have a default value.
   - True
   - False

2) The call CalcStat(44, 47, 42, 'b') uses usrMethod = 'a' because the parameter default value of 'a' overrides the argument 'b'.
   - True
   - False

3) The call CalcStat(44, 47, 42) uses usrMethod = 'a'.
   - True
   - False

4) The call CalcStat(44, 47, 'b') uses num3 = 0.
   - True
   - False

5) The following is a valid start of a function definition: `void myFct(int num1 = 0, int num2 = 0, char usrMethod) {`
   - True
   - False

Feedback?

Exploring further:

- Default arguments from msdn.microsoft.com

CHALLENGE ACTIVITY 5.12.1: Functions with default parameters.

620890.5010016.qx3zqy7

Start

Type the program's output

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

void PrintNums(int a, int b, int c = 15) {
   cout << a << ", " << b << ", " << c << endl;
}

int main() {
   PrintNums(2, 8, 1);
   PrintNums(3, 5);

   return 0;
}
```

```
2, 8, 1
3, 5, 15
```

1 2

Check Next

Feedback?

CHALLENGE ACTIVITY 5.12.2: Return number of pennies in total. Full screen

620890.5010016.qx3zqy7

Integer numDollars is read from input. Organize the lines of code to define a function NumberOfPennies() that returns the total number of pennies given the number of dollars and (optionally) a number of pennies.

Ex: If the input is 4, then the output is:

```
Balance: 400 pennies

Balance with quarter: 425 pennies
```

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

How to use this tool

Unused

```
}
int NumberOfPennies(int dollars, int pennies = 0) {
int NumberOfPennies(int dollars, int pennies) {
int NumberOfPennies(int dollars = 0, int pennies) {
return (dollars * 10) + pennies;
return (dollars * 100) + pennies
return (dollars * 100) + pennies;
```

main.cpp Load default template...

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

int main() {
   int numDollars;

   cin >> numDollars;

   cout << "Balance: " << NumberOfPennies(numDollars) << " pen
   cout << "Balance with quarter: " << NumberOfPennies(numDoll

   return 0;
}
```

Check

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH05.8-5.16 C5B
0 / 19 points
Due: 04/15/2025, 11:59 PM EDT
Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. Learn more

Completion details