6.12 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 6.12.1: Parameter with a default value.

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

// Function prints date in two styles (0: American (default), 1: European)
void DatePrint(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
   DatePrint(30, 7, 2012, 0);
   cout << endl;

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

   DatePrint(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 DatePrint(int currDay = 1, int currMonth = 1, int currYear = 2000, int pr`
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 DatePrint() function having default values for all parameters:

Figure 6.12.2: Valid function calls with default parameter values.

```
DatePrint(30, 7, 2012, 0); // No defaults
DatePrint(30, 7, 2012);    // Defaults:                          style=0
DatePrint(30, 7);          // Defaults:                 year=2000, style=0
DatePrint(30);             // Defaults:        month=1, year=2000, style=0 (strange, but valid)
DatePrint();               // 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 DatePrint(int currDay, int currMonth, int currYear, int printStyle = 0). Then the call DatePrint(30, 7) generates the following error message from g++.

Figure 6.12.3: 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 DatePrint(int, int, int, int)
fct_defparm.cpp:22: error: at this point in file
```

Feedback?

PARTICIPATION ACTIVITY 6.12.1: 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 6.12.1: Return number of pennies in total.

Write a function NumberOfPennies() that returns the total number of pennies given a number of dollars and (optionally) a number of pennies. Ex: 5 dollars and 6 pennies returns 506.

```
1 #include <iostream>
2 using namespace std;
3
```

```
4  /* Your solution goes here  */
5  int NumberOfPennies(int dollars, int pennies =0){
6     return dollars*100 + pennies;
7  }
8
9  int main() {
10    cout << NumberOfPennies(5, 6) << endl; // Should print 506
11    cout << NumberOfPennies(4) << endl;    // Should print 400
12    return 0;
13 }
```

Run ✓ All tests passed

✓ Testing with 5 dollars and 6 pennies

Your output `506`

✓ Testing with just 4 dollars

Your output `400`

Feedback?