Calculations Using Bloodshed Dev C++

By Janine Bouyssounouse

Bloodshed Dev C++ is a free program to make writing and compiling C++ programs easy to do. It makes executable files quickly so programs can be shared with others as soon as they are written and debugged. Bloodshed Dev C++ can be downloaded from the website: http://www.bloodshed.net/devcpp.html.

There is more than one way to start writing programs in Dev C++. This document discusses using a project for each program. This encourages using an organized file structure to keep each program in a separate location for future use. Remember that reusing code is a good idea and it is nice to be able to find the folder with everything for a program in one place.

Click on the File menu and select New – Project. Choose Console Application for this program. Type in a name for the project, such as Display Input, in the name field in the bottom left corner of the window. Click on the OK button. Choose a place on the computer to save the project. It's a good idea to make a folder for each project. Once the project is saved, some basic items show up on the screen. These items give a shell of a C++ program and include things that are unique to Bloodshed Dec C++ as well.

```
#include <iostream>       - This tells the compiler to include files.
#include <stdlib.h>       - This tells the compiler to include files.

using namespace std;      - This tells the compiler that you will be using key
                            words that are included in namespace std.

int main(int argc, char *argv[])        - This starts the main function.
{                                       - Main is contained between { and }.

  system("PAUSE");              - This leaves the console window open until
                                  you are ready to close it. Otherwise you
                                  wouldn't be able to see the results of the
                                  program. Not all compilers require this.
  return 0;                     - This is how the main program ends.
}                               - This signifies the end of the code for main.
```

The code for the main part of the program will be typed between the first curly brace ({) and the system("PAUSE") line of code. The input and print functions for this program will be outside of this area, but the main program will refer to them.

Most computer programs do some sort of mathematical calculations in them. They are involved in many ways to make the program work properly and most of the calculation results are not shown to the user on the screen. Addition (+), Subtraction (–), Multiplication (*) and Division (/) are the main operations, but there are higher math functions available, such as raising something to a power (^) and looking at the remainder of a division problem (%) that are common as well. Trigonometric functions are also available, but those are for a later date.

A computer can be used as an oversized calculator rather easily. Asking the user for input is the first step. Doing the calculations is the second step. The last step is displaying the information. Each of these steps can be done in individual functions.

Type in the first lines of the program by clicking at the beginning of the first line and pressing enter a couple of times, then moving to the first blank line. Type in these first two lines of code:

```
// This program gets input from the user, does calculations and displays
// the results using functions
```

The next line of code can be a comment stating your name as the programmer and the date the program was written.

```
// Written by Janine Bouyssounouse on 08/24/08
```

Type the following line of code after the using namespace statement:

```
int askNumber();
```

Int means there is an integer value returned to the main function from the askNumber function. AskNumber is the name of the function. The empty parentheses show that nothing is being passed to the askNumber function from the main program. The semicolon shows the end of the line of code.

On the last line of the program, we will comment the line to show that the main function is finished, so that it is not confused with the other functions listed after it.

The last line of code should look like this:

```
}   // end main
```

Next we will start typing the askNumber function at the end of the program, outside of the curly braces for the main function.

Skip a line and type:

```
// askNumber function asks for an integer from the user
// and passes it back to the main program
int askNumber()
```

Notice the comments are listed on the lines before the start of the function. The first line of the function looks exactly like the function prototype, except for the semicolon at the end.

On the next line of code, type in the function:

```
{
  int response;  // declares int variable response
  cout << "\nPlease type an integer: ";  // displays prompt to get input
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askNumber function
```

This function uses a variable to hold the input from the user. The variable name in the function is only for that function and a separate variable needs to be declared in the main function in order to handle the data returned from the askNumber function. I am using different variable names to illustrate the fact that they are actually different, even though they share the same information.

Declare the variable in the main function. Then call the askNumber function to assign the returned value to the newly declared variable.

To do this, type the following code after the first curly brace in the main function and before the system("PAUSE") line of code:

```
int num1, num2;  // declares two variables for user input

num1 = askNumber();  // calls askNumber and gives result to num1
num2 = askNumber();  // calls askNumber and gives result to num2
```

Note the declaration of two variables on the same line of code. This is optional and is up to the user. Grouping the variables for what they will be used for can make it easier to understand the code at a later date.

There is a need for two different variables with two different names because we want to do a calculation with two different numbers from the user. This shows the use of the same function to get more than one piece of information by calling it more than one time and using a different variable for each time it is called.

The program is only partially completed. We now have input from the user, but we still need to do something with that input. Since the next function needs to have the information from the first function, we are going to pass that information to the next function. Now the parentheses at the end of the function name will be filled, instead of empty.

Type the following code after the askNumber function prototype at the beginning of the program:

```
int add(int num1, int num2);
```

Int states there will be and integer returned from the function. Int num1, int num2 in the parentheses states there will be two integer variables passed to the function from the main function. I used the same variable titles (num1, num2) as used in the main function, but they can have different titles.

At the end of the program, type in the add function:

```
// add function gets two integers from the main function, adds
// them together and returns the result of the calculation
// back to the main function
int add(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 + num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end add function
```

The variables passed do not need to be declared inside of the function as the last one did. They are passed and used with the title given to them in the first line of the function.

A new variable is created to store the answer to the calculation. The two passed variables are added together and the result is passed back to the main function.

Before calling the add function, another variable needs to be declared in the main function to hold the result of the add function. Declare the new variable on the line following the num1 and num2 variable declaration in the main function.

```
int answer;  // declares answer variable for the calculation result
```

Now the new function needs to be called from the main function with the information gathered in the askNumber function. Type the following code into the main function after the askNumber function has been called the second time:

```
answer = add(num1, num2); // passes num1 and num2 to add function
```

The num1 and num2 variables are in the parentheses after the name of the function as a way to get the contents of the variables to the function, so the function can do the calculations.

Now that the input has been gathered and the calculation has been done, the results need to be displayed to the user. It is a nice idea to show the completed information that shows the user input as well as the result of the

calculation. This means all three variables need to be passed to the display function.

Another function prototype needs to be added after the two listed at the top of the program.

```
void display(int num1, int num2, int answer);
```

The same variable names are used as in the main function to help understand what order the variables should be entered when calling the function. This is a void function because nothing will be returned to the main function.

Next the function needs to be added at the end of the program.

```
// display function gets three integers from the main function
// and displays them to the user on the screen
void display(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " + " << num2 << " = " << answer << "\n\n"; // the result
  return;
}  // end display function
```

The new function needs to be called from main or else it will not be used in the program. Type in the following line after the add function is called in the main function.

```
display (num1, num2, answer);  // passes variables to display function
```

The program is finished. Here is the code:

```
// This program gets input from the user, does calculations and displays
// the results using functions
// Written by Janine Bouyssounouse on 08/24/08

#include <iostream>
#include <stdlib.h>

using namespace std;
```

```
int askNumber();
int add(int num1, int num2);
void display(int num1, int num2, int answer);

int main(int argc, char *argv[])
{
  int num1, num2;  // declares two variables for user input
  int answer;  // declares answer variable for the calculation result

  num1 = askNumber();  // calls askNumber and gives result to num1
  num2 = askNumber();  // calls askNumber and gives result to num2
  answer = add(num1, num2); // passes num1 and num2 to add function
  display (num1, num2, answer);  // passes variables to display function

  system("PAUSE");
  return 0;
}  // end main

// askNumber function asks for an integer from the user
// and passes it back to the main function
int askNumber()
{
  int response;  // declares int variable response
  cout << "\nPlease type an integer: ";  // displays prompt to get input
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askNumber function

// add function gets two integers from the main function, adds
// them together and returns the result of the calculation
// back to the main function
int add(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 + num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end add function

// display function gets three integers from the main function
```

```
// and displays them to the user on the screen
void display(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " + " << num2 << " = " << answer << "\n\n"; // the result
  return;
}  // end display function
```

Save, compile and run the program to see if it works. Choose Compile and Run from the Execute menu.

Here is a display of the program:

```
Please type an integer: 6

Please type an integer: 7


The calculation is done.

6 + 7 = 13

Press any key to continue . . .
```

This program waited for input on the first line. Once the user typed the input and pressed the enter key, the second line of code printed.

Now write a program of your own to write your own input, calculation and print functions.

Exercise 1: Write a program with an input function asking the user for two numbers, then pass this information to two calculation functions and pass that information to two print functions to display the results of the calculations. Choose names for the functions that tell what they do.

Exercise 2: Write a program with and input, four calculation functions and four print functions and call them each from the main function. Use the % modulus operator to give the remainder for the division problem. This can be done in a fifth calculation function.

Exercise 3: Write a program that calculates a base raised to a power, such as $3^2 = 3 \times 3 = 9$. 3 is the base and 2 is the power (exponent). Be sure to include <cmath> at the beginning of the program and use the pow function, which takes two doubles (one for the base and one for the power) and returns a double. Example: answer = pow(base, power) where answer, base and power are all declared as double variables instead of int variables.

Exercise 4: Write a program to calculate the square root of a number, such as $\sqrt{9} = 3$. The square root of 9 is 3. Be sure to include <cmath> at the beginning of the program to use the sqrt function, which takes a double and returns a double. Example: answer = sqrt(number) where answer is the square root of number.

Sample Code for Exercise 1:

```
// This program gets input from the user, does calculations and displays
// the results using functions
// Written by Janine Bouyssounouse on 08/24/08

#include <iostream>
#include <stdlib.h>

using namespace std;

int askNumber();
int add(int num1, int num2);
int multiply(int num1, int num2);
void displayAdd(int num1, int num2, int answerAdd);
void displayMult(int num1, int num2, int answerMult);

int main(int argc, char *argv[])
{
  int num1, num2;  // declares two variables for user input
  int answerAdd;  // declares answerAdd variable for the calculation result
  int answerMult;  // declares answerMult variable for the calculation result

  num1 = askNumber();  // calls askNumber and gives result to num1
  num2 = askNumber();  // calls askNumber and gives result to num2
  answerAdd = add(num1, num2); // passes num1 and num2 to add function
  answerMult = multiply(num1, num2); // passes num1, num2 to multiply
function
  displayAdd (num1, num2, answerAdd);  // passes variables to displayAdd
  displayMult (num1, num2, answerMult); // passes variables to displayMult

  system("PAUSE");
  return 0;
}  // end main

// askNumber function asks for an integer from the user
// and passes it back to the main function
int askNumber()
{
  int response;  // declares int variable response
```

```
  cout << "\nPlease type an integer: ";  // displays prompt to get input
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askNumber function

// add function gets two integers from the main function, adds
// them together and returns the result of the calculation
// back to the main function
int add(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 + num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end add function

// multiply function gets two integers from the main function, multiplies
// them together and returns the result of the calculation
// back to the main function
int multiply(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 * num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end multiply function

// displayAdd function gets three integers from the main function
// and displays them to the user on the screen
void displayAdd(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " + " << num2 << " = " << answer << "\n\n"; // the result
  return;
}  // end displayAdd function

// displayMult function gets three integers from the main function
// and displays them to the user on the screen
void displayMult(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " x " << num2 << " = " << answer << "\n\n"; // the result
```

```
  return;
}  // end displayMult function
```

Display from Sample Code for Exercise 1:

```
Please type an integer: 3

Please type an integer: 7


The calculation is done.

3 + 7 = 10



The calculation is done.

3 x 7 = 21

Press any key to continue . . .
```

Sample Code for Exercise 2:

```
// This program gets input from the user, does calculations and displays
// the results using functions
// Written by Janine Bouyssounouse on 08/24/08

#include <iostream>
#include <stdlib.h>

using namespace std;

int askNumber();
int add(int num1, int num2);
int multiply(int num1, int num2);
int subtract(int num1, int num2);
int divide(int num1, int num2);
```

```
int mod(int num1, int num2);
void displayAdd(int num1, int num2, int answerAdd);
void displayMult(int num1, int num2, int answerMult);
void displaySubt(int num1, int num2, int answerSubt);
void displayDiv(int num1, int num2, int answerDiv, int answerMod);

int main(int argc, char *argv[])
{
  int num1, num2;  // declares two variables for user input
  int answerAdd;  // declares answerAdd variable for the calculation result
  int answerMult;  // declares answerMult variable for the calculation result
  int answerSubt; // declares answerSubt variable for the calculation result
  int answerDiv;  // declares answerDiv variable for the caculation result
  int answerMod;  // declares answerMod variable for the division remainder

  num1 = askNumber();  // calls askNumber and gives result to num1
  num2 = askNumber();  // calls askNumber and gives result to num2
  answerAdd = add(num1, num2); // passes num1 and num2 to add function
  answerMult = multiply(num1, num2); // passes num1, num2 to multiply
function
  answerSubt = subtract(num1, num2); // passes num1, num2 to subtract
function
  answerDiv = divide(num1, num2);  // passes num1, num2 to divide
function
  answerMod = mod(num1, num2); // passes num1, num2 to mod function
for remainder
  displayAdd(num1, num2, answerAdd);  // passes variables to displayAdd
  displayMult(num1, num2, answerMult); // passes variables to displayMult
  displaySubt(num1, num2, answerSubt); // passes variables to displaySubt
  displayDiv(num1, num2, answerDiv, answerMod); // calls displayDiv

  system("PAUSE");
  return 0;
}  // end main

// askNumber function asks for an integer from the user
// and passes it back to the main function
int askNumber()
{
  int response;  // declares int variable response
```

```
  cout << "\nPlease type an integer: ";  // displays prompt to get input
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askNumber function

// add function gets two integers from the main function, adds
// them together and returns the result of the calculation
// back to the main function
int add(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 + num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end add function

// multiply function gets two integers from the main function, multiplies
// them together and returns the result of the calculation
// back to the main function
int multiply(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 * num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end multiply function

// subtract function gets two integers from the main function, multiplies
// them together and returns the result of the calculation
// back to the main function
int subtract(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 - num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end subtract function

// divide function gets two integers from the main function, multiplies
// them together and returns the result of the calculation
// back to the main function
int divide(int num1, int num2)
{
```

```
  int answer;  // declares integer variable answer
  answer = num1 / num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end divide function

// mod function gets two integers from the main function, multiplies
// them together and returns the result of the calculation
// back to the main function
int mod(int num1, int num2)
{
  int answer;  // declares integer variable answer
  answer = num1 % num2;  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end mod function

// displayAdd function gets three integers from the main function
// and displays them to the user on the screen
void displayAdd(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " + " << num2 << " = " << answer << "\n\n"; // the result
  return;
}  // end displayAdd function

// displayMult function gets three integers from the main function
// and displays them to the user on the screen
void displayMult(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " x " << num2 << " = " << answer << "\n\n"; // the result
  return;
}  // end displayMult function

// displaySubt function gets three integers from the main function
// and displays them to the user on the screen
void displaySubt(int num1, int num2, int answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " - " << num2 << " = " << answer << "\n\n"; // the result
  return;
```

```
}  // end displaySubt function

// displayDiv function gets three integers from the main function
// and displays them to the user on the screen
void displayDiv(int num1, int num2, int answerDiv, int answerMod)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << num1 << " / " << num2 << " = " << answerDiv;
  cout << " with a remainder of " << answerMod << "\n\n"; // the result
  return;
}  // end displayDiv function
```

Display from Sample Code for Exercise 2:

```
Please type an integer: 20

Please type an integer: 3


The calculation is done.

20 + 3 = 23



The calculation is done.

20 x 3 = 60



The calculation is done.

20 - 3 = 17



The calculation is done.
```

```
20 / 3 = 6 with a remainder of 2

Press any key to continue . . .
```

Sample Code for Exercise 3:

```
// This program gets input from the user, does calculations and displays
// the results using functions
// Written by Janine Bouyssounouse on 01/06/09

#include <iostream>
#include <stdlib.h>
#include <cmath>

using namespace std;

void welcome();
double askBase();
double askPower();
double calculate(double base, double power);
void displayAnswer(double base, double power, double answer);

int main(int argc, char *argv[])
{
  double base, power;  // declares two variables for user input
  double result;  // declares answer variable for the calculation result

  welcome();  // calls welcome function
  base = askBase();  // calls askBase and gives result to base
  power = askPower();  // calls askPower and gives result to power
  result = calculate(base, power); // passes base and power to calcualte
function
  displayAnswer(base, power, result);  // passes variables to displayAnswer

  system("PAUSE");
  return 0;
}  // end main

// welcome function displays an explanation of the program
```

```
void welcome()
{
  cout << "This program will ask for two numbers. One will be the\n";
  cout << "base and the other will be the power the base will be\n";
  cout << "raised to in the calculation. The answer will then be\n";
  cout << "displayed on the screen.\n\n\n";
}  // end welcome function

// askBase function asks for an integer from the user
// and passes it back to the main function
double askBase()
{
  double response;  // declares variable response
  cout << "\nPlease type a number for the base: ";  // displays prompt
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askBase function

// askPower function asks for a number from the user
// and passes it back to the main function
double askPower()
{
  double response;  // declares variable response
  cout << "\nPlease type a number for the power: ";  // displays prompt
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askPower function

// calculate function gets two numbers from the main function,
// raises the base to the power and returns the result of the calculation
// back to the main function
double calculate(double base, double power)
{
  double answer;  // declares variable answer
  answer = pow(base, power);  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end calculate function

// displayAnswer function gets three numbers from the main function
// and displays them to the user on the screen
```

```
void displayAnswer(double base, double power, double answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << base << " raised to the power of " << power ;
  cout << " is " << answer << "\n\n";
  return;
}  // end displayAnswer function
```

Display from Sample Code for Example 3:

```
This program will ask for two numbers. One will be the
base and the other will be the power the base will be
raised to in the calculation. The answer will then be
displayed on the screen.



Please type a number for the base: 3

Please type a number for the power: 4


The calculation is done.

3 raised to the power of 4 is 81

Press any key to continue . . .
```

Sample Code for Exercise 4:

```
// This program gets input from the user, does calculations and displays
// the results using functions
// Written by Janine Bouyssounouse on 01/21/09

#include <iostream>
#include <stdlib.h>
#include <cmath>

using namespace std;
```

```
void welcome();
double askNumber();
double calculate(double number);
void displayAnswer(double number, double answer);

int main(int argc, char *argv[])
{
  double number;  // declares two variables for user input
  double result;  // declares answer variable for the calculation result

  welcome();  // calls welcome function
  number = askNumber();  // calls askNumber and gives result to number
  result = calculate(number); // passes number to calcualte function
  displayAnswer(number, result);  // passes variables to displayAnswer

  system("PAUSE");
  return 0;
}  // end main

// welcome function displays an explanation of the program
void welcome()
{
  cout << "This program will ask for a number. Then the program\n";
  cout << "will calculate the square root of that number\n";
  cout << "and display the result on the screen.\n\n\n";
}  // end welcome function

// askNumber function asks for a number from the user
// and passes it back to the main function
double askNumber()
{
  double response;  // declares int variable response
  cout << "\nPlease type a number: ";  // displays prompt
  cin >> response;  // places the user input into response variable
  return response;  // sends the contents of response back to main
}  // end of askNumber function

// calculate function gets a number from the main function
// and calculates the square root of that number, then sends
```

```
// the square root back to the main function
double calculate(double number)
{
  double answer;  // declares variable answer
  answer = sqrt(number);  // does calculation and puts result into answer
  return answer;  // returns the result to main
}  // end calculate function

// displayAnswer function gets two numbers from the main function
// and displays them to the user on the screen
void displayAnswer(double number, double answer)
{
  cout << "\n\nThe calculation is done.\n\n";
  cout << "The square root of " << number;
  cout << " is " << answer << ".\n\n";
  return;
}  // end displayAnswer function
```

Display from Sample Code for Exercise 4:

```
This program will ask for a number. Then the program
will calculate the square root of that number
and display the result on the screen.



Please type a number: 121


The calculation is done.

The square root of 121 is 11.

Press any key to continue . . .
```