

Display Input 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 Dev 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.

Involving the end user of the program makes a program interactive. Most computer programs ask for input from the user and then do something with the input and then display the results. So it is important to learn how to obtain information from the user to use in a program and then how to display information based on what the user input.

Making the input statement be in a function opens up the concept of how to pass information back and forth between the main program and the functions. So far the print functions have not needed any information to do their job. Now we need to get information from the user and then pass that information to the print function to display it on the screen.

Type in the first line 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 this first line of code:

```
// This program gets input from the user and displays it using functions
```

The second 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 06/22/08
```

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

```
string askName();
```

String means there is a string value returned to the main function from the askName function. AskName is the name of the function. The empty parentheses show that nothing is being passed to the askName 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 askName function at the end of the program, outside of the curly braces for the main function.

Skip a line and type:

```
// askName function asks for the user's name  
// and passes it back to the main program  
string askName()
```

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:

```
{  
    string answer; // declares string variable answer  
    cout << "Please type your name: "; // displays question to get input  
    cin >> answer; // places user response into answer variable  
    return answer; // sends the contents of answer back to main  
} // end of askName 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 askName 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 askName 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:

```
string name; // declares string variable name

name = askName(); // calls askName function and puts it in name variable
```

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 `askName` function prototype at the beginning of the program:

```
void displayName(string name);
```

Void states there will be no value returned from the function. String `name` in the parentheses states there will be a string variable passed to the function from the main function. I used the same variable title (`name`) as used in the main function, but it can have a different title.

At the end of the program, type in the `displayName` function:

```
// displayName function receives the contents of a name variable
// and then displays a message with the name variable information
void displayName(string name)
{
    cout << "\n\nWelcome to programming, " << name << "! \n\n";
    return;
} // end of displayName function
```

The variable does not need to be declared inside of the function as the last one did. It is passed and used with the title given to it in the first line of the function.

The cout statement has multiple items listed separately. The << marks indicate a different element is being used in the statement. Listing the name of the variable in the cout statement displays it on the screen.

Now the new function needs to be called from the main function with the information gathered in the askName function. Type the following code into the main function after the askName function:

```
displayName(name); // calls displayName with contents of name variable
```

The name variable is in the parentheses after the name of the function as a way to get the contents of the variable to the function, so the function can print it to the screen.

The program is finished. Here is the code:

```
//This program gets input from the user and displays it using functions
// Written by Janine Bouyssounouse 06/22/08

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

using namespace std;

string askName();
void displayName(string name);

int main(int argc, char *argv[])
{
    string name; // declares string variable name

    name = askName(); // calls askName function and puts it in name variable
    displayName(name); // calls displayName with contents of name variable

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

// askName function asks for the user's name
```

```
// and passes it back to the main program
string askName()
{
    string answer; // declares string variable answer
    cout << "Please type your name: "; // displays question to get input
    cin >> answer; // places user response into answer variable
    return answer; // sends the contents of answer back to main
} // end of askName function

// displayName function receives the contents of a name variable
// and then displays a message with the name variable information
void displayName(string name)
{
    cout << "\n\nWelcome to programming, " << name << "! \n\n";
    return;
} // end of displayName 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 your name: Janine

Welcome to programming, Janine!

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 and print functions.

Exercise 1: Write a program with an input function asking the user what her favorite color is, then pass this information to the print function to display the user's favorite color. Choose names for the functions that tell what they do.

Exercise 2: Write a program with an input and two print functions and call them each from the main function. Write one function to print a welcome message, then ask for information from the user, then display a message to the user using the information given.

Exercise 3: Write a program to ask the user for his or her name. Then print this name on the screen three times on three different lines.

Sample Code for Exercise 1:

```
//This program gets input from the user and displays it using functions
// Written by Janine Bouyssounouse 06/28/08

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

using namespace std;

string askColor();
void displayColor(string color);

int main(int argc, char *argv[])
{
    string color; // declares string variable name

    color = askColor(); // calls askColor function and puts it in name variable
    displayColor(color); // calls displayColor with contents of name variable

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

// askColor function asks for the user's favorite color
// and passes it back to the main program
string askColor()
{
    string answer; // declares string variable answer
    cout << "What is your favorite color? "; // displays question to get input
    cin >> answer; // places user response into answer variable
    return answer; // sends the contents of answer back to main
} // end of askColor function

// displayColor function receives the contents of a color variable
// and then displays a message with the color variable information
void displayColor(string color)
{
    cout << "\n\nI'm so glad your favorite color is " << color << "! \n\n";
    return;
}
```

```
} // end of displayColor function
```

Display from Sample Code for Exercise 1:

```
What is your favorite color? yellow
```

```
I'm so glad your favorite color is yellow!
```

```
Press any key to continue . . .
```

Sample Code for Exercise 2:

```
//This program gets input from the user and displays it using functions
// Written by Janine Bouyssounouse 06/28/08

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

using namespace std;

void welcome();
string askSeason();
void displaySeason(string season);

int main(int argc, char *argv[])
{
    string season; // declares string variable name

    welcome(); // this calls the welcome function
    season = askSeason(); // calls askSeason function and puts it in season
variable
    displaySeason(season); // calls displaySeason with contents of season
variable

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

```

void welcome() // intro text displayed at beginning of program
{
    cout << "\tWelcome to my program!\n\n";
    cout << "This program asks for your favorite season.";
    cout << " Then it displays that season.\n\n";
} // end welcome function

// askSeason function asks for the user's favorite season
// and passes it back to the main program
string askSeason()
{
    string answer; // declares string variable answer
    cout << "What is your favorite season? "; // displays question to get input
    cin >> answer; // places user response into answer variable
    return answer; // sends the contents of answer back to main
} // end of askSeason function

// displaySeason function receives the contents of a season variable
// and then displays a message with the season variable information
void displaySeason(string season)
{
    cout << "\n\nHey, " << season << " is my favorite season, too!\n\n";
    return;
} // end of displaySeason function

```

Display from Sample Code for Exercise 2:

```

Welcome to my program!

This program asks for your favorite season. Then it displays that season.

What is your favorite season? winter

Hey, winter is my favorite season, too!

Press any key to continue . . .

```

Sample Code for Exercise 3:

```
//This program gets input from the user and displays it using functions
// Written by Janine Bouyssounouse 01/21/09

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

using namespace std;

string askName();
void displayName(string name);

int main(int argc, char *argv[])
{
    string name; // declares string variable name

    name = askName(); // calls askName function and puts it in name variable
    displayName(name); // calls displayName with contents of name variable

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

// askName function asks for the user's favorite season
// and passes it back to the main program
string askName()
{
    string answer; // declares string variable answer
    cout << "What is your name? "; // displays question to get input
    cin >> answer; // places user response into answer variable
    return answer; // sends the contents of answer back to main
} // end of askName function

// displayName function receives the contents of a name variable
// and then displays a message with the name variable information
void displayName(string name)
{
    cout << "\n\n" << name << "\n\t" << name << "\n" << name << "\n\n";
}
```

```
return;  
} // end of displayName function
```

Display from Sample Code for Exercise 3:

```
What is your name? Janine  
  
Janine  
    Janine  
Janine  
  
Press any key to continue . . .
```