Makara Ramoabi

20240045

C++

1.**Scenario Analysis**

Creating a program that greets the user is quite easy in context. It involves the greeting, the name input (Name of the user), and the #include <string>. It starts of quite simple like creating any regular program, with a #include <iostream> and then a #include <string>. Then 'using namespace std;'. On the next line of code, we use int main() {. This starts the instruction, meaning this is where the program starts to be executed. Next line we use 'string name;' then continue down the plage with 'cout<<"Enter Name"; '. getline(cin, name);. On the next line is 'cout<<"Hello"<<name<<"!"<<endl;, return 0. Then end the instructions with "}". On the work space it would look like this:

```
#include <iostream>

#include <string>

Using namespace std;

Int main () {

String name

Cout<<" Enter Name";

getline (cin, name);

cout<<" Hello" <<name<<"!" <<endl;

return 0

}
```

This is a simple and useful for wed design, or any other user based application.

2.**Component research**

One important program component in C++ is the **namespace.** A namespace is used to organize code and avoid name conflicts between identifiers such as variables, functions, and classes. It allows programmers to group related code together, making large programs easier to manage and

read. The most commonly used namespace in C++ is the **std**, which contains features like input (cin) and output (cout), and other standard library functions. To use it, programmers can write using '**using namespace std;'** at the beginning of their code. Without namespaces, programs with multiple libraries could have conflicting variable or function names, leading to errors and confusion.

3.**Tool practice**

Creating a temperature converter is a bit complex to get running, it starts of normal like all programs. It starts of with **#include <iostream>**, then move on to the next line of to write **using namespacing std;**. On the next line we start with **int main () {**, this starts with instructions of the programs. On the next line of code is "double Celsius, Fahrenheit;", we move on to the next lines and write;

Cout<<"Temperature in Celsius:";

Cin>>Celsius;

Fahrenheit=(Celsius *9/5)+32;

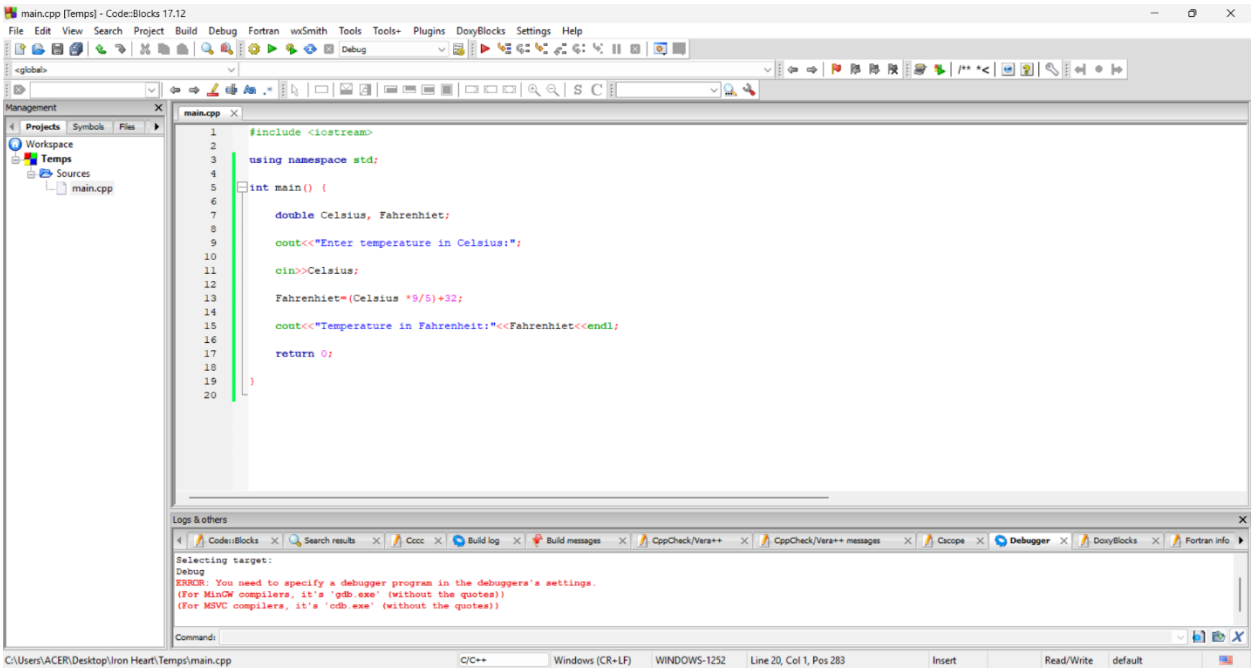
Cout<<" Temperature in Fahrenheit"<<Fahrenheit<<endl;

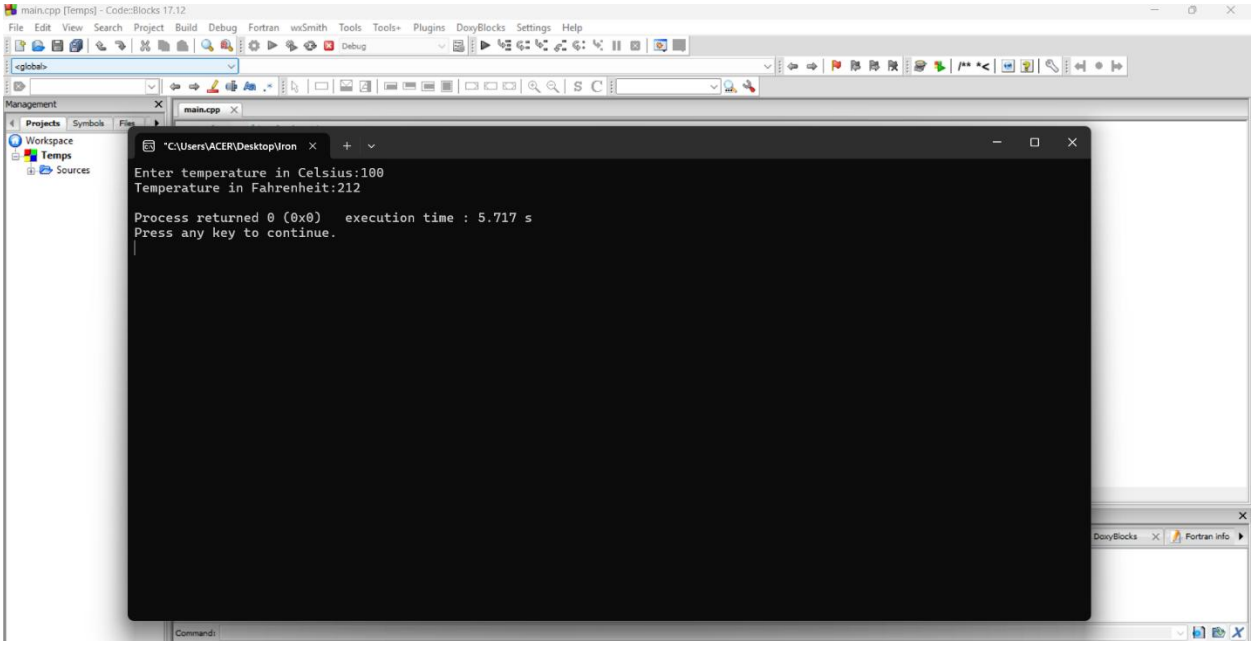
Return 0;

}

My biggest challenge was debugging the code, particularly the spelling of "Celsius &

Fahrenheit". I commonly misspelled both words, changing the first character from capital letter to a non-capital.


main.cpp [Temps] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
Temps
Sources
main.cpp
main.cpp
#include <iostream>
using namespace std;
int main() {
double Celsius, Fahrenheit;
cout<<"Enter temperature in Celsius:";
cin>>Celsius;
Fahrenheit=(Celsius *9/5)+32;
cout<<"Temperature in Fahrenheit:"<<Fahrenheit<<endl;
return 0;
}
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
Fortran info
Selecting target:
Debug
ERROR: You need to specify a debugger program in the debuggers's settings.
(For MinGW compilers, it's 'gdb.exe' (without the quotes))
(For MSVC compilers, it's 'cdb.exe' (without the quotes))
Command:
C:\Users\ACER\Desktop\Iron Heart\Temps\main.cpp
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 20, Col 1, Pos 283
Insert
Read/Write
default


main.cpp [Temps] - Code::Blocks 17.12
"C:\Users\ACER\Desktop\Iron
Enter temperature in Celsius:100
Temperature in Fahrenheit:212
Process returned 0 (0x0) execution time : 5.717 s
Press any key to continue.

4.