

Laboratory Report 6

Familiarization with TurboC++ IDE

V. QUESTIONS

A. Navigating the IDE

1. Menu Bar Exploration

- What options are available under the "File" menu? Describe the purpose of each option.

New: Creates a new, blank document or project. This is often used to start fresh without affecting existing files.

Open: Opens an existing document or file from your storage. This allows you to view or edit files that you've previously created or saved.

Save: Saves the current document or project, updating the existing file with any changes made since the last save.

Save as: Allows you to save the current document under a new name or in a different format/location. This is useful for creating copies or versions of a file.

Save all: Saves all open documents or projects at once, ensuring that any unsaved changes in multiple files are stored simultaneously.

Change directory: Changes the current working directory or folder where files are accessed or saved. This is particularly useful for organizing files or navigating to a specific location.

Print: Sends the current document or project to a printer, allowing you to create a physical copy. This often includes options for print settings like page range, number of copies, etc.

DOS Shell: Opens a command-line interface (if supported), allowing users to run DOS commands. This is useful for advanced users needing to execute scripts or commands directly.

Quit: Closes the application. If there are unsaved changes, the application usually prompts you to save before exiting.

- How do you access the compiler in TurboC++? Which menu options are involved?

To access the compiler in Turbo C++, open the IDE, create or open a source file from the File menu, write your code, then compile it by selecting Compile from the Compile menu (or pressing Alt + F9), and finally run the program from the Run menu (or by pressing Ctrl + F9).

2. Editor Window Functions

- How do you save a C++ program in TurboC++? What file extension should be used?

To save a C++ program in Turbo C++, go to the File menu and select Save or Save As, then enter your desired file name followed by the .cpp extension (e.g., program.cpp). This extension indicates that it is a C++ source file.

- What keyboard shortcuts can you use for copy, paste, and undo in the editor?

Copy: Ctrl + Ins
Paste: Shift + Ins
Undo: Alt + Backspace

3. Compiling and Running

- What steps are involved in compiling and running a program in TurboC++?

Open Turbo C++: Launch the Turbo C++ IDE.
Create or Open a Source File: Use File > New to create a new file or File > Open to open an existing one.
Write Your Code: Enter your C++ code in the editor window.
Save Your Program: Save the file using File > Save (or Save As) with a .cpp extension.
Compile the Code: Go to the Compile menu and select Compile (or press Alt + F9) to check for syntax errors.
Run the Program: If there are no errors, go to the Run menu and select Run (or press Ctrl + F9) to execute the program.
View Output: The output will be displayed in the output window.

- What happens if there is an error in your code during the compilation process? How does TurboC++ notify you of errors?

If there's an error during compilation in Turbo C++, the IDE displays an error message in the output window, indicating the line number and type of error, allowing you to locate and fix it before recompiling.

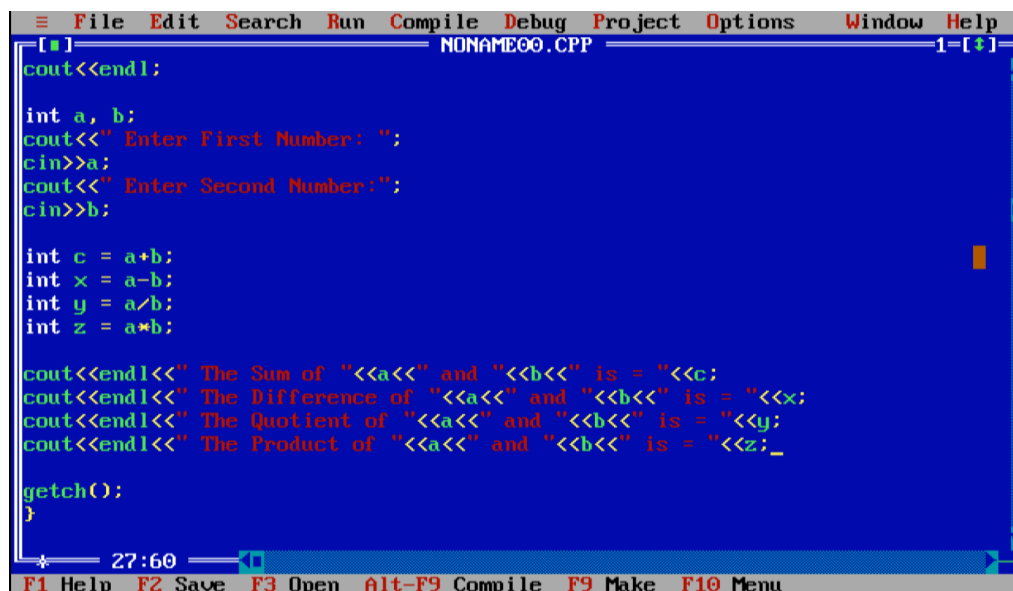
B. Practical Application

1. Writing a Program

- Write a simple program that prompts the user to enter two numbers and displays their sum, difference, quotient and product. Save, compile, and run the program in TurboC++.

Take a screenshot of the code and the output, then paste it here.

Code:



```
File Edit Search Run Compile Debug Project Options Window Help
[.] NONAME00.CPP 1-[+]
```

```
cout<<endl;

int a, b;
cout<<" Enter First Number: ";
cin>>a;
cout<<" Enter Second Number:";
cin>>b;

int c = a+b;
int x = a-b;
int y = a/b;
int z = a*b;

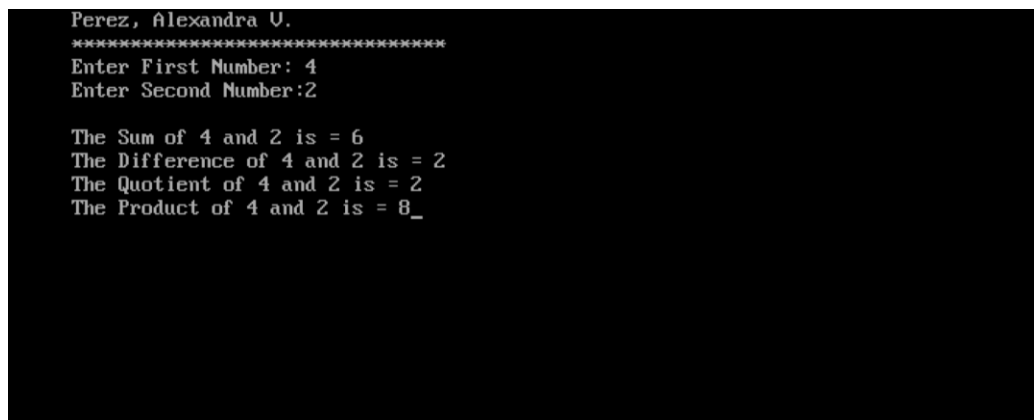
cout<<endl<<" The Sum of "<<a<<" and "<<b<<" is = "<<c;
cout<<endl<<" The Difference of "<<a<<" and "<<b<<" is = "<<x;
cout<<endl<<" The Quotient of "<<a<<" and "<<b<<" is = "<<y;
cout<<endl<<" The Product of "<<a<<" and "<<b<<" is = "<<z;_

getch();
}
```

27:60

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

Result:



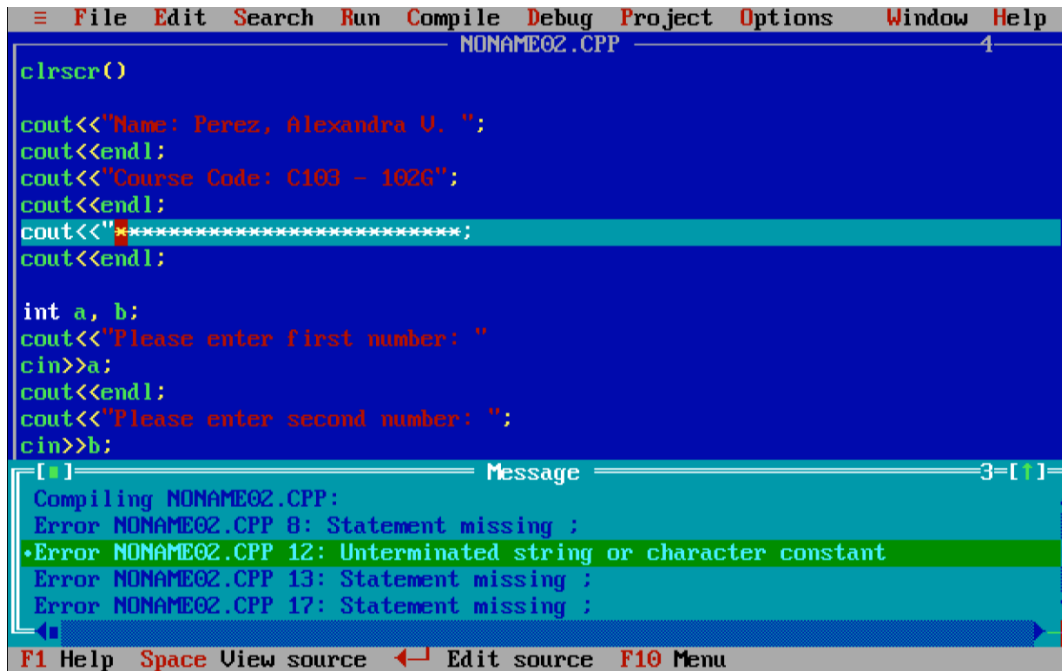
```
Perez, Alexandra V.
*****
Enter First Number: 4
Enter Second Number:2

The Sum of 4 and 2 is = 6
The Difference of 4 and 2 is = 2
The Quotient of 4 and 2 is = 2
The Product of 4 and 2 is = 8_
```

2. Debugging

- Introduce an error into your program (e.g., forget to include a semicolon). Describe the error message displayed by TurboC++ and the process of debugging the error. Take a screenshot of the code and the output, then paste it here.

Code:



The screenshot shows the TurboC++ IDE with a C++ program in a file named NONAME02.CPP. The code includes a clrscr() function call, several cout statements for displaying text, and two cin statements for user input. There are three syntax errors highlighted in the Message window: 'Error NONAME02.CPP 8: Statement missing ;', 'Error NONAME02.CPP 12: Unterminated string or character constant', and 'Error NONAME02.CPP 13: Statement missing ;'. The code in the editor is as follows:

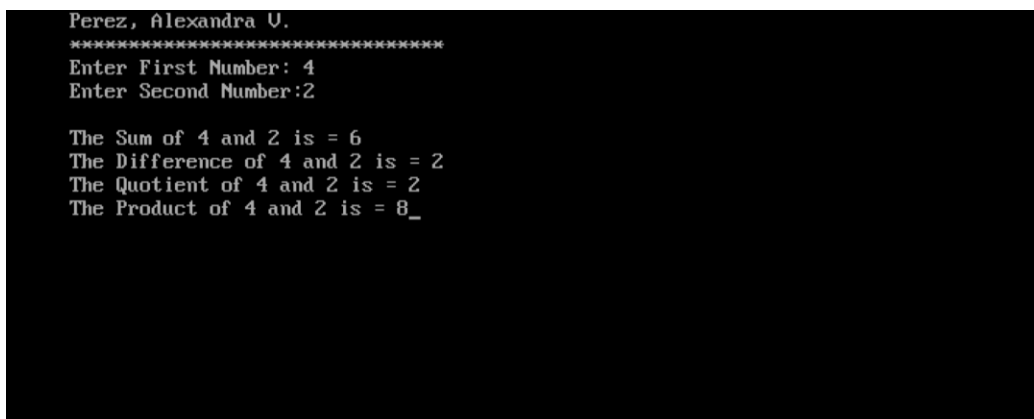
```
clrscr()

cout<<"Name: Perez, Alexandra V. ";
cout<<endl;
cout<<"Course Code: C103 - 102G";
cout<<endl;
cout<<"*****";
cout<<endl;

int a, b;
cout<<"Please enter first number: ";
cin>>a;
cout<<endl;
cout<<"Please enter second number: ";
cin>>b;
```

After running the code, you may notice a **Syntax error**. The compiler will indicate the line where the error occurred and provide details about the error. You can then debug your code and run it again until all errors are resolved.

Result:



The screenshot shows the output of the program after the errors have been corrected. The output is as follows:

```
Perez, Alexandra V.
*****
Enter First Number: 4
Enter Second Number:2

The Sum of 4 and 2 is = 6
The Difference of 4 and 2 is = 2
The Quotient of 4 and 2 is = 2
The Product of 4 and 2 is = 8_
```

VI. CONCLUSION

Becoming familiar with an IDE like Turbo C++ is essential for C++ programming, as it simplifies tasks like code editing, compiling, and debugging. Mastering Turbo C++ enhances your efficiency and confidence, making it easier to tackle future programming challenges and adapt to other development environments. This foundational skill can greatly benefit your career in technology by preparing you for more advanced tools and languages.