PASCAL PROGRAMMING LANGUAGE (1)

CMP 102
Introduction to computer programming
(Module 7A)

Dr. S. A. Arekete

A First Look at a Pascal Program

- Following is a simple Pascal code that would print the words "Hello, World!":

```
program HelloWorld;
uses crt;
(* Here the main program block starts *)
begin
   writeln('Hello, World!');
   readkey;
end.
```

Pascal Programming Language

- Pascal is a general-purpose, high-level language that was originally developed by Niklaus Wirth in the early 1970s.
- It was developed for teaching programming as a systematic discipline and to develop reliable and efficient programs.
- Pascal is an Algol-based language and includes many constructs of Algol.
- Algol-60 is a subset of Pascal.
- Pascal offers several data types and programming structures.
- It is easy to understand and maintain the Pascal programs.

Why Pascal is popular in Academics Arena

- Pascal has grown in popularity in the teaching and academics arena for various reasons:
  - Easy to learn.
  - Structured language.
  - It produces transparent, efficient and reliable programs.
  - It can be compiled on a variety of computer platforms.

Features of the Pascal Language

- Pascal has the following features:
  - Pascal is a strongly typed language.
  - It offers extensive error checking.
  - It offers several data types like arrays, records, files and sets.
  - It offers a variety of programming structures.
  - It supports structured programming through functions and procedures.
  - It supports object oriented programming.

Facts about Pascal

- The Pascal language was named for Blaise Pascal, French mathematician and pioneer in computer development.
- Niklaus Wirth completed development of the original Pascal programming language in 1970.
- Pascal is based on the block structured style of the Algol programming language.
- Pascal was developed as a language suitable for teaching programming as a systematic discipline, whose implementations could be both reliable and efficient.

Facts about Pascal..

- The ISO 7185 Pascal Standard was originally published in 1983.
- Pascal was the primary high-level language used for development in the Apple Lisa, and in the early years of the Mac.
- In 1986, Apple Computer released the first Object Pascal implementation, and in 1993, the Pascal Standards Committee published an Object-Oriented Extension to Pascal.

Why using Pascal?

- Pascal allows the programmers to define complex structured data types and build dynamic and recursive data structures, such as lists, trees and graphs.
- Pascal offers features like records, enumerations, subranges, dynamically allocated variables with associated pointers and sets.
- Pascal allows nested procedure definitions to any level of depth.
  - This truly provides a great programming environment for learning programming as a systematic discipline based on the fundamental concepts.

Practical Programs Written in Pascal

- Among the most amazing programs implemented in Pascal are:
  - Skype
  - Total Commander
  - TeX
  - Macromedia Captivate
  - Apple Lisa
  - Various PC Games
  - Embedded Systems

Pascal Environments

- There are several Pascal compilers and interpreters available for general use.
- Among these are:
  - Turbo Pascal: provides an IDE and compiler for running Pascal programs on CP/M, CP/M-86, DOS, Windows and Macintosh.
  - Delphi: provides compilers for running Object Pascal and generates native code for 32- and 64-bit Windows operating systems, as well as 32-bit Mac OS X and iOS. Embarcadero is planning to build support for the Linux and Android operating system.
  - Free Pascal: it is a free compiler for running Pascal and Object Pascal programs. Free Pascal compiler is a 32- and 64-bit Turbo Pascal and Delphi compatible Pascal compiler for Linux, Windows, OS/2, FreeBSD, Mac OS X, DOS and several other platforms.
  - Turbo51: it is a free Pascal compiler for the 8051 family of microcontrollers, with Turbo Pascal 7 syntax.
  - Oxygene: it is an Object Pascal compiler for the .NET and Mono platforms.
  - GNU Pascal (GPC): it is a Pascal compiler composed of a front end to GNU Compiler Collection.

Installing Free Pascal on Windows

- For Windows, you will download the Windows installer, setup.exe.
- This is a usual installation program.
- You need to take the following steps for installation:
  - Select a directory.
  - Select parts of the package you want to install.
  - Optionally choose to associate the .pp or .pas extensions with the Free Pascal IDE.
  - We will be using Free Pascal in this course. You can download Free Pascal for your operating system from the link:

http://www.freepascal.org/download.var

Text Editor

- This will be used to type your program.
- Examples of few editors include:
- Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.
- Name and version of text editor can vary on different operating systems.
  - For example, Notepad will be used on Windows and vim or vi can be used on Windows as well as Linux or UNIX.
- The files you create with your editor are called source files and contain program source code.
- The source files for Pascal programs are typically named with the extension .pas.
- Before starting your programming, make sure you have one text editor in place and you have enough experience to write a computer program, save it in a file, compile it and finally execute it.
- It is good to note that Freepascal has an in-built editor, it is an integrated development environment (IDE) from were you can create and run your program.

Program Structure or Template

- A Pascal program basically consists of the following parts:
  - Program name
  - Uses command
  - Type declarations
  - Constant declarations
  - Variables declarations
  - Functions declarations
  - Procedures declarations
  - Main program block
  - Statements and Expressions within each block
  - Comments

Program Structure..

- Every Pascal program generally have a heading statement, a declaration and an execution part strictly in that order.
- Following format shows the basic syntax for a Pascal program:

Program Structure..

```
program {name of the program}
uses {comma delimited names of libraries you use}
const {global constant declaration block}
var {global variable declaration block}
function {function declarations, if any}
{ local variables }
begin
...
end;
procedure { procedure declarations, if any}
{ local variables }
begin
...
end;
begin { main program block starts}
...
end. { the end of main program block }
```

Pascal Hello World Example

- Following is a simple Pascal code that would print the words "Hello, World!":

```
program HelloWorld;
uses crt;
(* Here the main program block starts *)
begin
   writeln('Hello, World!');
   readkey;
end.
```

Pascal Hello World Example..

- Let us look various parts of the above program:
  - The first line of the program program HelloWorld; indicates the name of the program.
  - The second line of the program uses crt; is a preprocessor command, which tells the compiler to include the crt unit before going to actual compilation.
  - The next lines enclosed within begin and end statements are the main program block.
    - Every block in Pascal is enclosed within a begin statement and an end statement.
    - However, the end statement indicating the end of the main program is followed by a full stop (.) instead of semicolon (;).
  - The begin statement of the main program block is where the program execution begins.

Pascal Hello World Example..

- The lines within (*...*) will be ignored by the compiler and it has been put to add a comment in the program.
- The statement writeln('Hello, World!'); uses the writeln function available in Pascal which causes the message "Hello, World!" to be displayed on the screen.
- The statement readkey; allows the display to pause until the user presses a key. It is part of the crt unit. A unit is like a library in Pascal.
- The last statement end. ends your program.

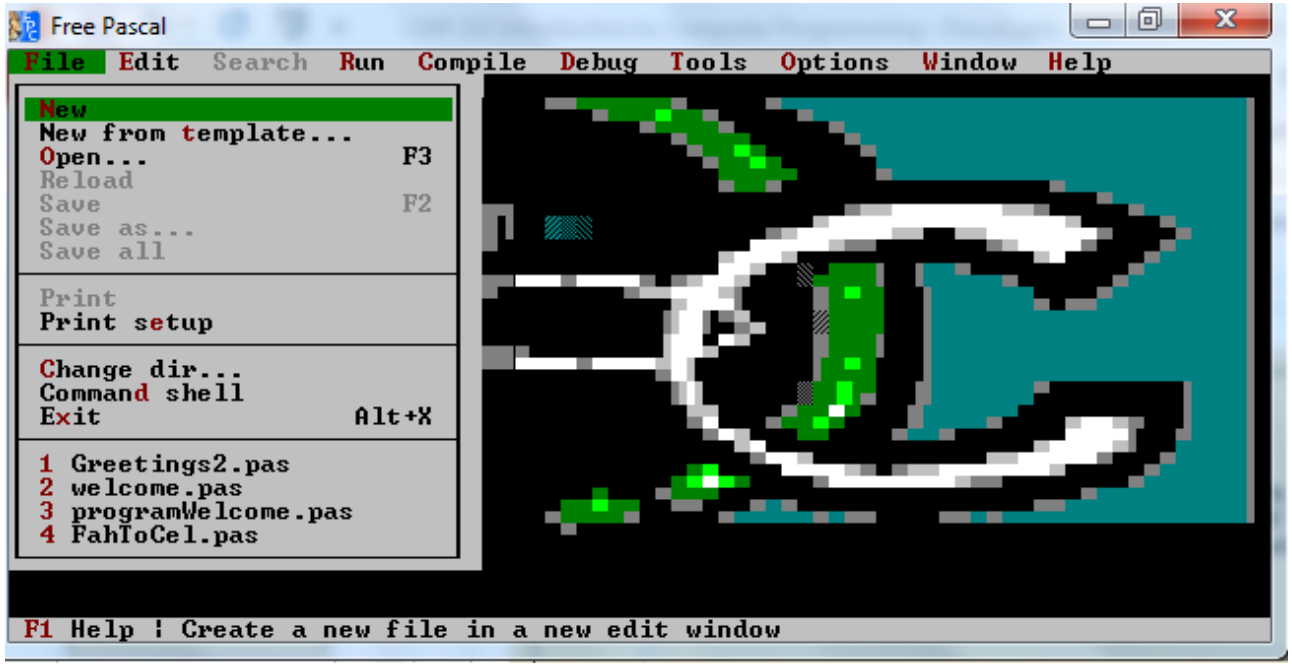
Compiling and Executing a Pascal Program (the General Method)

- Open a text editor and add the above-mentioned code.
- Save the file as hello.pas
- Open a command prompt and go to the directory, where you saved the file.
- Type fpc hello.pas at command prompt and press enter to compile your code.
- If there are no errors in your code, the command prompt will take you to the next line and would generate **hello** executable file and **hello.o** object file.
- Now, type **hello** at command prompt to execute your program.
- You will be able to see "Hello World" printed on the screen and program waits till you press any key.

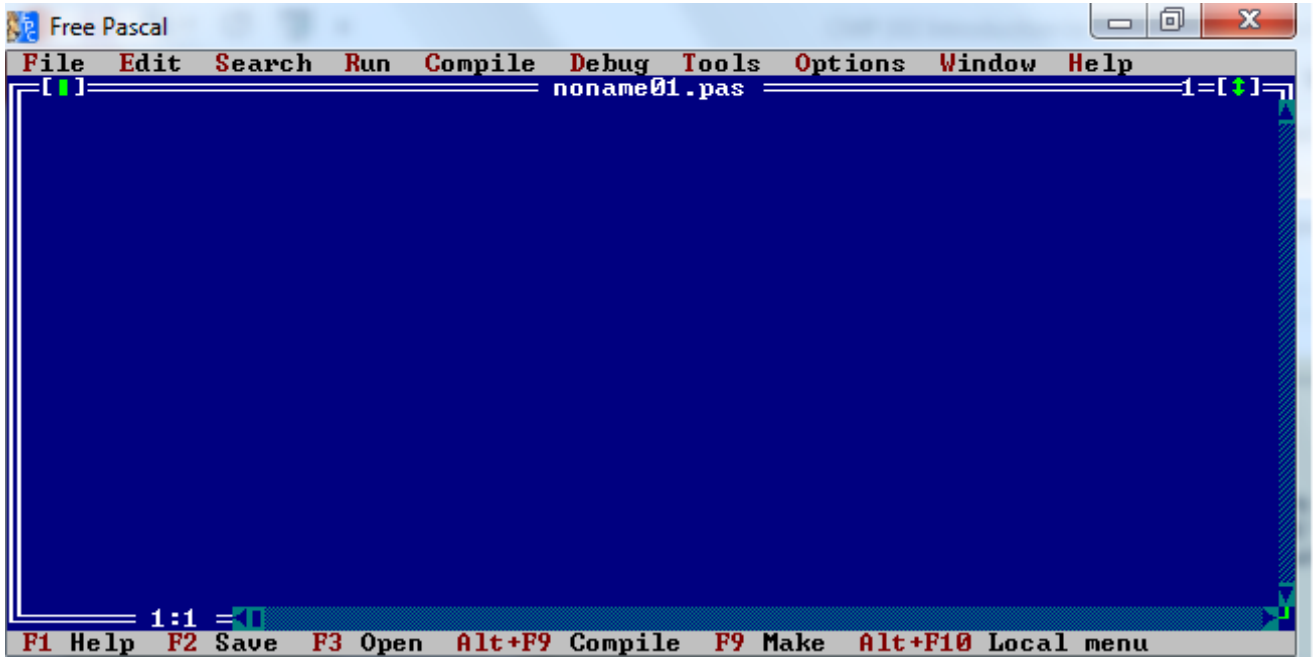
Pascal IDE

- The Integrated Development Environment (IDE) which is common nowadays makes writing, compiling and executing a program much easier.
- The editor and the compiler are all embedded inside the same environment.
- The in-built editor is used to edit the program.
- Compile and RUN commands can be invoked on the menu to compile the program and execute it.

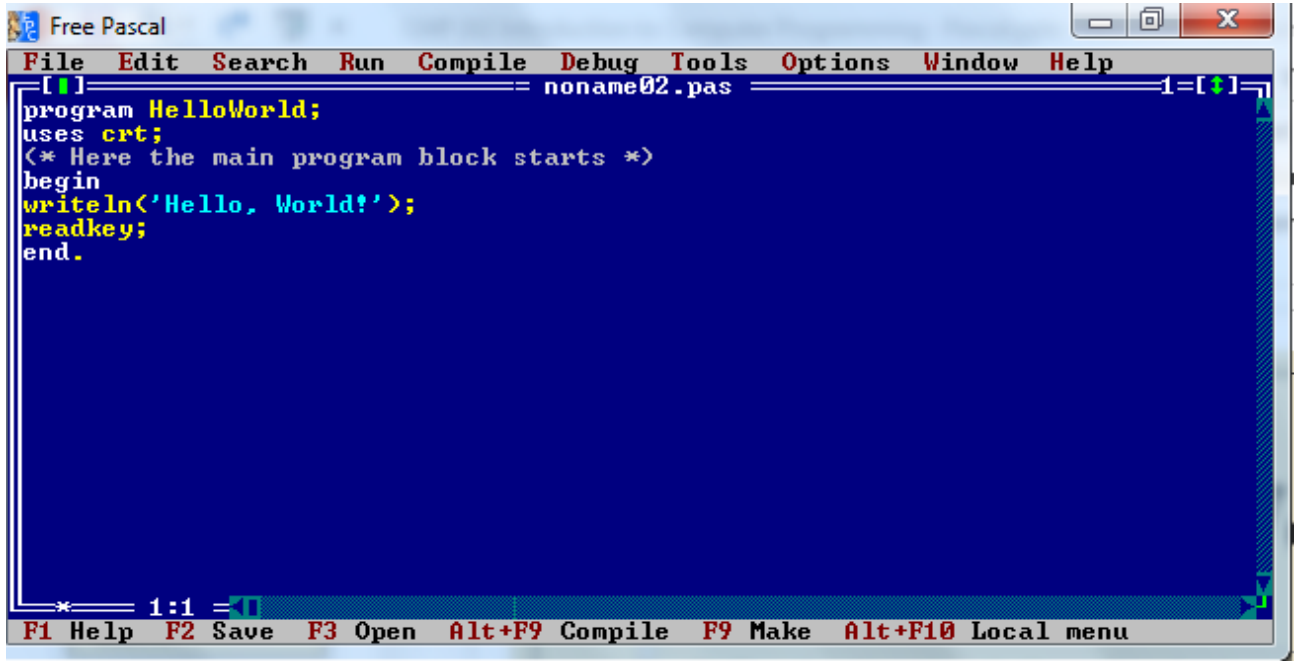
Pascal IDE..


Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
New
New from template...
Open... F3
Reload
Save F2
Save as...
Save all
Print
Print setup
Change dir...
Command shell
Exit Alt+X
1 Greetings2.pas
2 welcome.pas
3 programWelcome.pas
4 FahToCel.pas
F1 Help | Create a new file in a new edit window

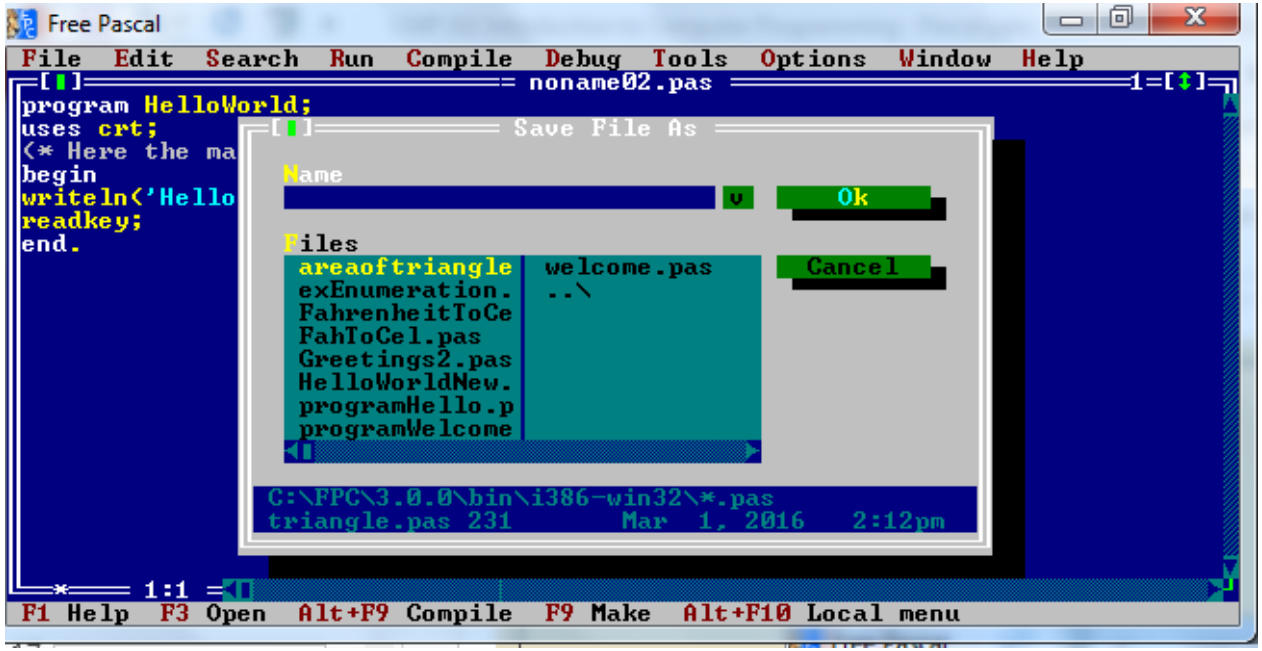
Pascal IDE..


Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
noname01.pas
1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

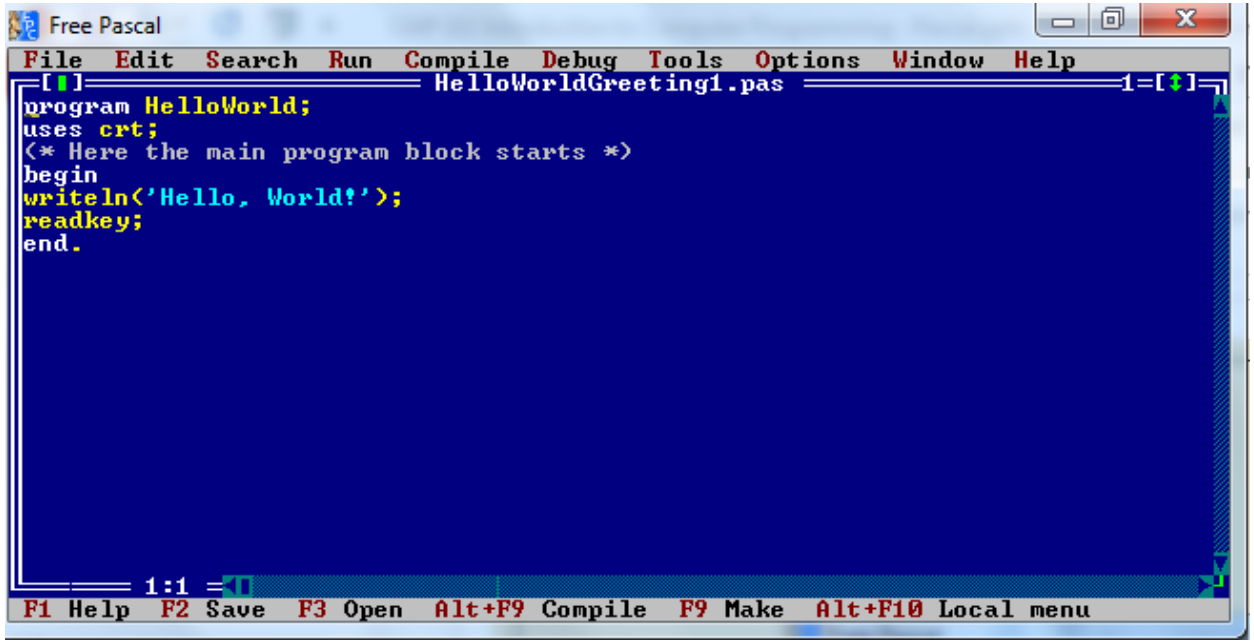
Pascal IDE..


Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
noname02.pas
program HelloWorld;
uses crt;
(* Here the main program block starts *)
begin
writeln('Hello, World!');
readkey;
end.
1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

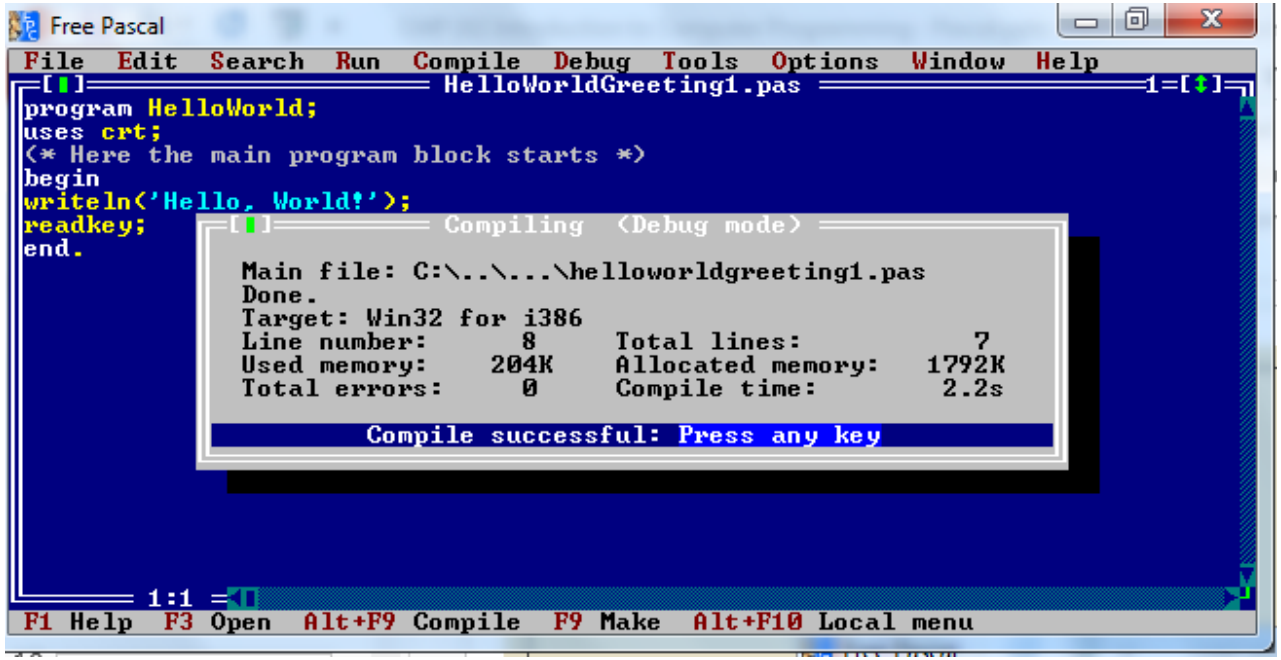
Pascal IDE..


Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
noname02.pas
program HelloWorld;
uses crt;
(* Here the ma
begin
writeln('Hello
readkey;
end.
Save File As
Name
Ok
Files
areaoftriangle
exEnumeration.
FahrenheitToCe
FahToCel.pas
Greetings2.pas
HelloWorldNew.
programHello.p
programWelcome
welcome.pas
..\
Cancel
C:\FPC\3.0.0\bin\i386-win32*.pas
triangle.pas 231 Mar 1, 2016 2:12pm
1:1
F1 Help F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

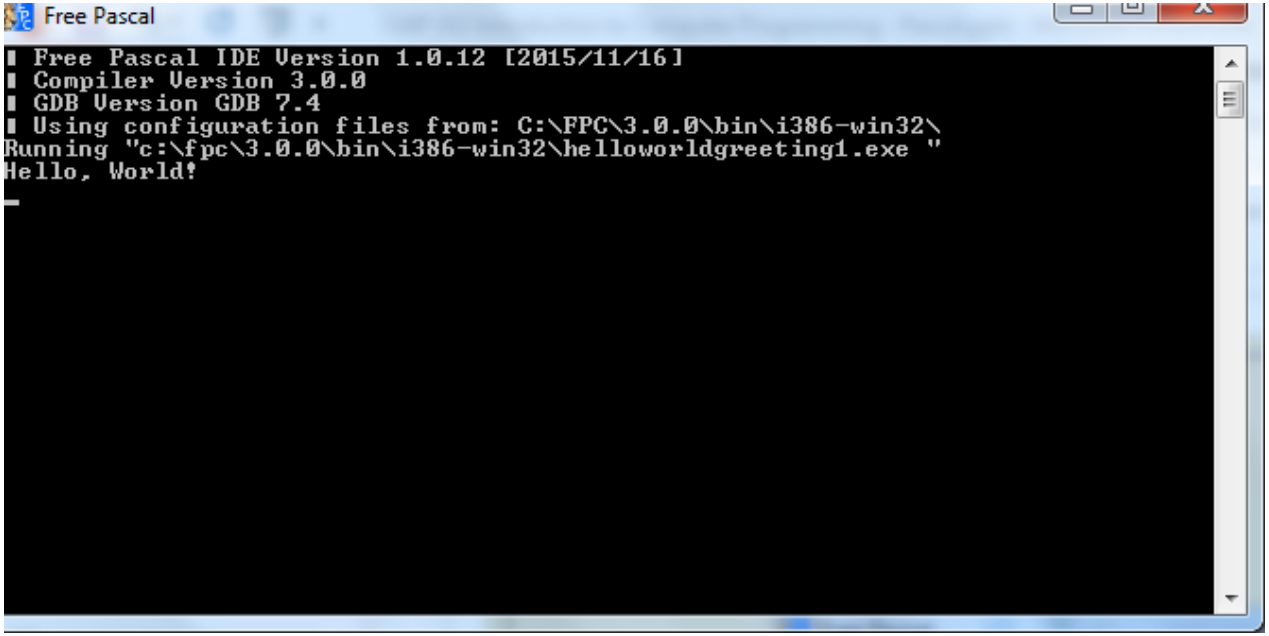
Pascal IDE..


Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
HelloWorldGreeting1.pas
program HelloWorld;
uses crt;
(* Here the main program block starts *)
begin
writeln('Hello, World!');
readkey;
end.
1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Pascal IDE..


Free Pascal
File Edit Search Run Compile Debug Tools Options Window Help
HelloWorldGreeting1.pas
program HelloWorld;
uses crt;
(* Here the main program block starts *)
begin
writeln('Hello, World!');
readkey;
end.
Compiling (Debug mode)
Main file: C:\..\...\helloworldgreeting1.pas
Done.
Target: Win32 for i386
Line number: 8 Total lines: 7
Used memory: 204K Allocated memory: 1792K
Total errors: 0 Compile time: 2.2s
Compile successful: Press any key
1:1
F1 Help F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Pascal IDE..


Free Pascal
Free Pascal IDE Version 1.0.12 [2015/11/16]
Compiler Version 3.0.0
GDB Version GDB 7.4
Using configuration files from: C:\FPC\3.0.0\bin\i386-win32\
Running "c:\fpc\3.0.0\bin\i386-win32\helloworldgreeting1.exe "
Hello, World!