

# IDES E PROGRAMAÇÃO

---

Prof. Dr. Cláudio Fabiano Motta Toledo

PAE: Maurício A Dias

{claudio,macdias}@icmc.usp.br

# Agenda

- Conceitos
- Instalação e Configuração
- Compilação e Execução de Código
- IDEs
- Exemplo
- Projetos – Criação e Compilação

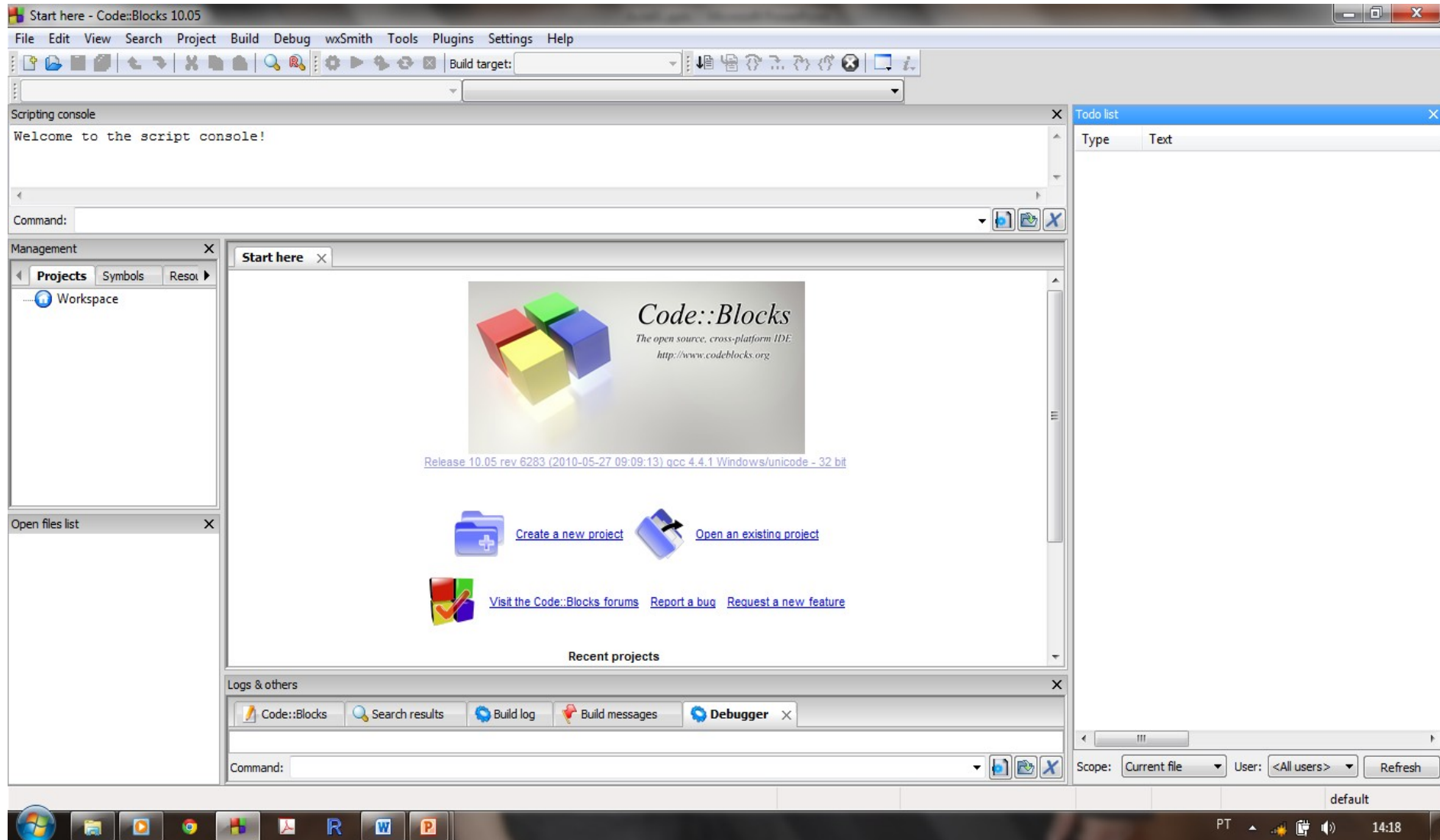
# Conceitos

- O que é uma IDE?
  - *Integrated Development Environment*
  - Software que agrupa funcionalidades
    - No caso, software que agrupa funcionalidades de programação
- Vantagens
  - Compilação Transparente
  - Criação de Projetos
  - Configurações Básicas Transparentes
  - Editor de texto com várias características interessantes
  - As IDEs permitem que pacotes sejam instalados automaticamente para desenvolvimento de aplicações complexas que utilizam bibliotecas de outros desenvolvedores

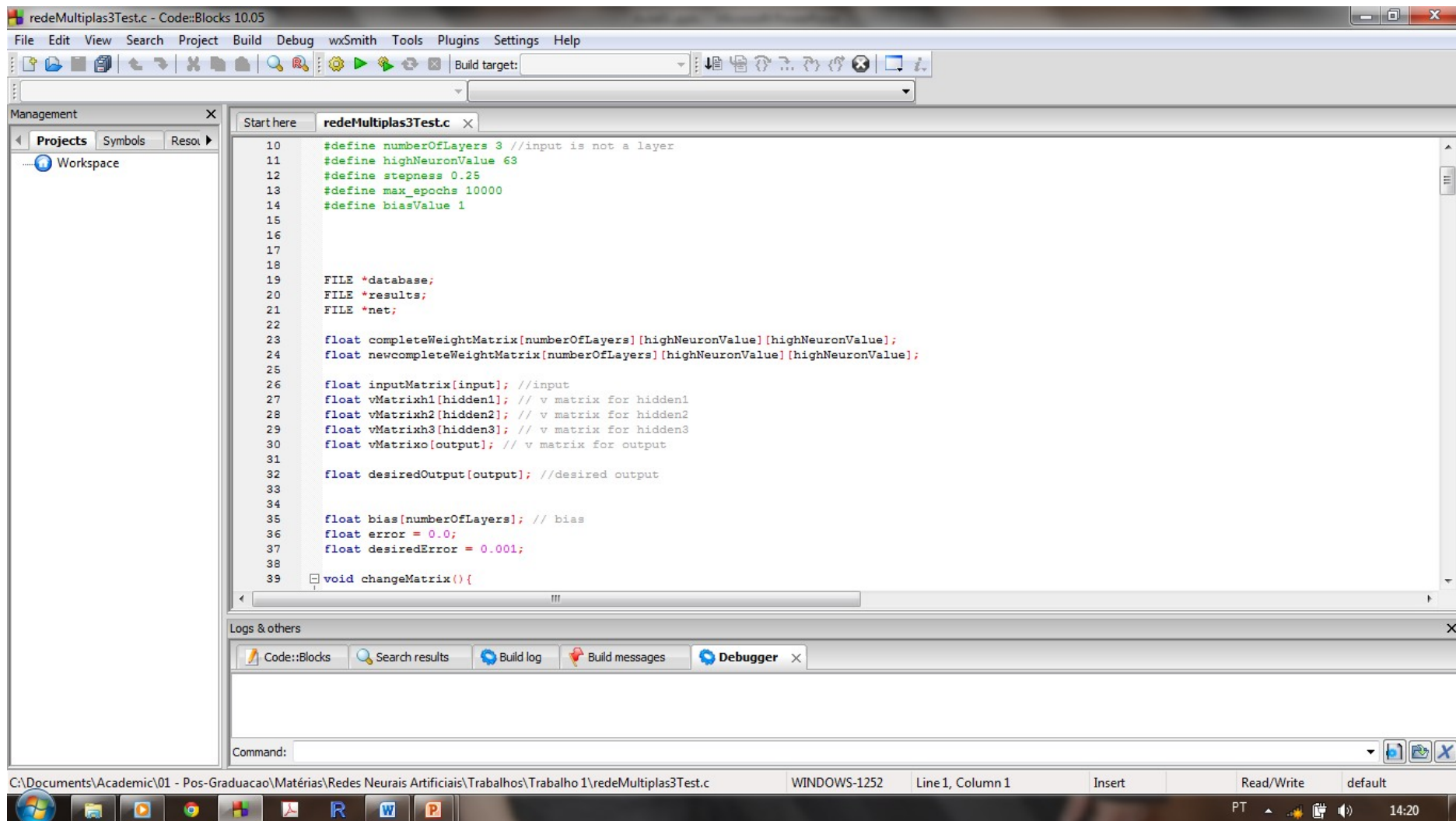
# Conceitos

- Desvantagens
  - Aparentemente a programação em IDEs não apresenta desvantagens, porém o usuário pode se acostumar com a transparência criando desta maneira uma “dependência”

# Conceitos



# Conceitos



The image shows a screenshot of a code editor window titled "redeMultiplas3Test.c - Code::Blocks 10.05". The editor displays C code for a neural network test. The code includes preprocessor directives for defining constants, file inclusions, and variable declarations. The code is as follows:

```
10 #define numberOfLayers 3 //input is not a layer
11 #define highNeuronValue 63
12 #define stepness 0.25
13 #define max_epochs 10000
14 #define biasValue 1
15
16
17
18
19 FILE *database;
20 FILE *results;
21 FILE *net;
22
23 float completeWeightMatrix[numberOfLayers][highNeuronValue][highNeuronValue];
24 float newcompleteWeightMatrix[numberOfLayers][highNeuronValue][highNeuronValue];
25
26 float inputMatrix[input]; //input
27 float vMatrixh1[hidden1]; // v matrix for hidden1
28 float vMatrixh2[hidden2]; // v matrix for hidden2
29 float vMatrixh3[hidden3]; // v matrix for hidden3
30 float vMatrixo[output]; // v matrix for output
31
32 float desiredOutput[output]; //desired output
33
34
35 float bias[numberOfLayers]; // bias
36 float error = 0.0;
37 float desiredError = 0.001;
38
39 void changeMatrix(){
```

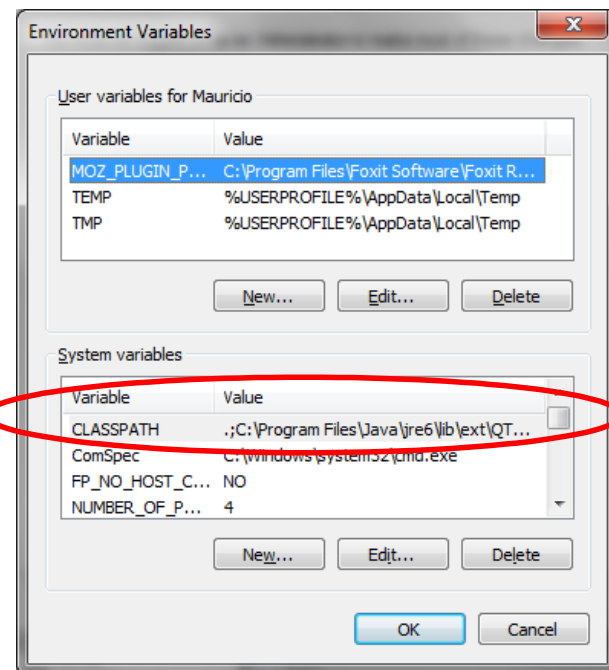
The code editor interface includes a menu bar (File, Edit, View, Search, Project, Build, Debug, wxSmith, Tools, Plugins, Settings, Help), a toolbar, a Management panel on the left with tabs for Projects, Symbols, and Resou, and a Logs & others panel at the bottom with tabs for Code::Blocks, Search results, Build log, Build messages, and Debugger. The status bar at the bottom shows the file path "C:\Documents\Academic\01 - Pos-Graduacao\Matérias\Redes Neurais Artificiais\Trabalhos\Trabalho 1\redeMultiplas3Test.c", the window title "WINDOWS-1252", and the cursor position "Line 1, Column 1". The system tray at the bottom right shows the time "14:20".

# Instalação e Configuração

- Linux
  - Instalação utilizando aptget ou gerenciador de pacotes por exemplo
  - As variáveis de ambiente são configuradas automaticamente
- Windows
  - A IDE possui um instalador no site do desenvolvedor
    - Fazer o download e instalar
  - Compilador
    - Caso o compilador não venha integrado é necessário baixar o compilador e indicar o caminho para a IDE (dificilmente acontece)
    - Para utilizar o compilador sem a IDE é necessário configurar a variável de ambiente para que o prompt de comando saiba onde estão os arquivos do compilador

# Instalação e Configuração

- No caso do compilador C/C++ para windows, sua versão é o MinGW. Este compilador é gratuito e está disponível para download. Após instalar acessar:
  - Computador (botão direito) -> Propriedades -> Aba Advanced possui um botão chamado Variáveis de Ambiente
  - Computer (right button) -> Properties -> Advanced button Environment Variables
- Ao clicar aparece uma tela como esta:
  - Clicar 2x em path localizado mais abaixo.
  - Não confundir com CLASSPATH

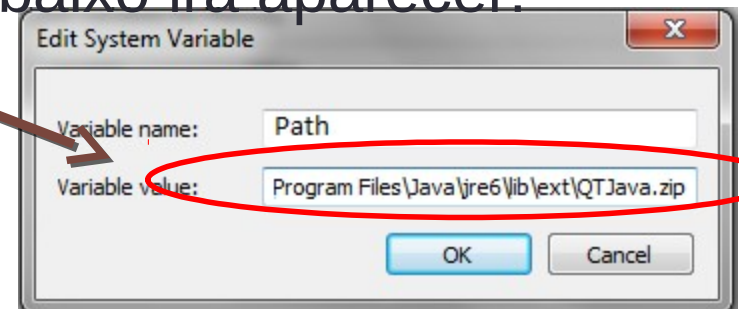




# Instalação e Configuração

- Uma janela semelhante à janela abaixo irá aparecer:

- Editar o valor da variável



- Para editar o valor:

- acrescente um ; ao final.
- Cole o endereço da pasta de binários do seu compilador

(provavelmente chama-se 'bin') em seguida, sem espaço entre o ; e o começo do endereço.

- Não acrescente ; ao final do último endereço
- Clique OK

# Instalação e Configuração

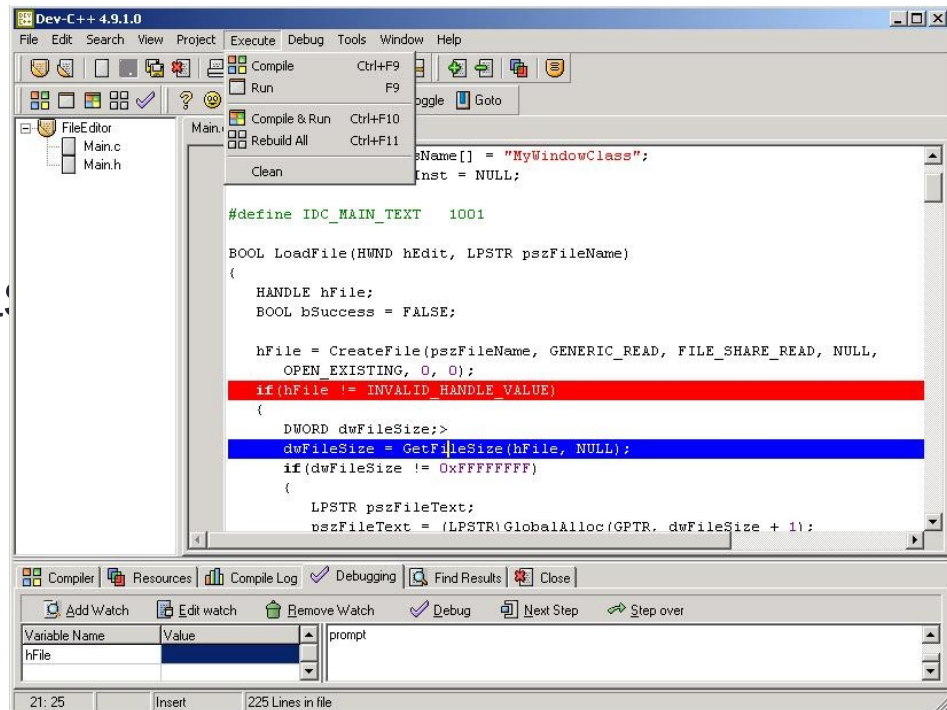
- Pronto a configuração está feita. Agora é possível também utilizar o compilador e seus programas diretamente no prompt de comando. O teste pode ser feito digitando-se `gcc` no prompt de comando e *enter* em seguida.
- Irá aparecer a mensagem
  - `gcc: no input files`
- Caso sua IDE já possua o compilador integrado é so colar o caminho dos binários do compilador que deverá estar na pasta referente à IDE criada no diretório Arquivos de Programas ou *Program Files*.

# IDEs

- Existem várias IDEs de programação disponíveis no mercado, algumas delas são
  - Dev c++
  - Code::Blocks
  - Microsoft Visual Studio
  - Netbeans
  - Notepad++
- O importante é notar que independente da plataforma utilizada o código limpo, indentado e comentado é extremamente recomendável

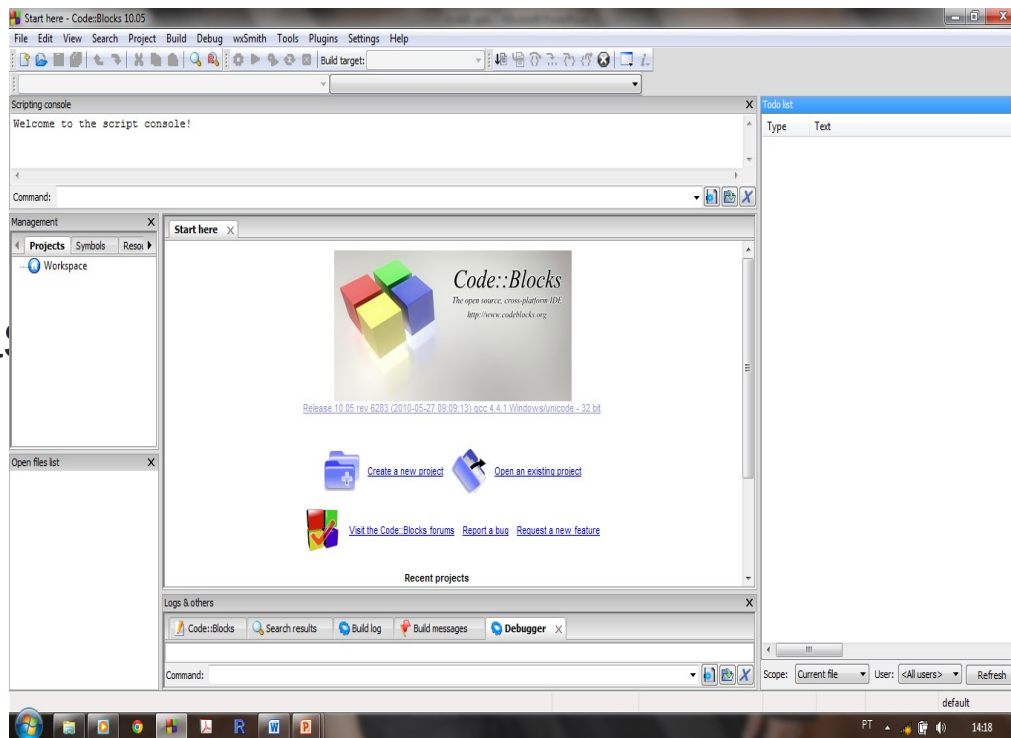
# IDEs

- Existem várias IDEs de mercado, algumas delas
  - Dev c++
  - Code::Blocks
  - Microsoft Visual Studio
  - Netbeans
  - Notepad++
- O importante é notar que independente da plataforma utilizada o código limpo, indentado e comentado é extremamente recomendável



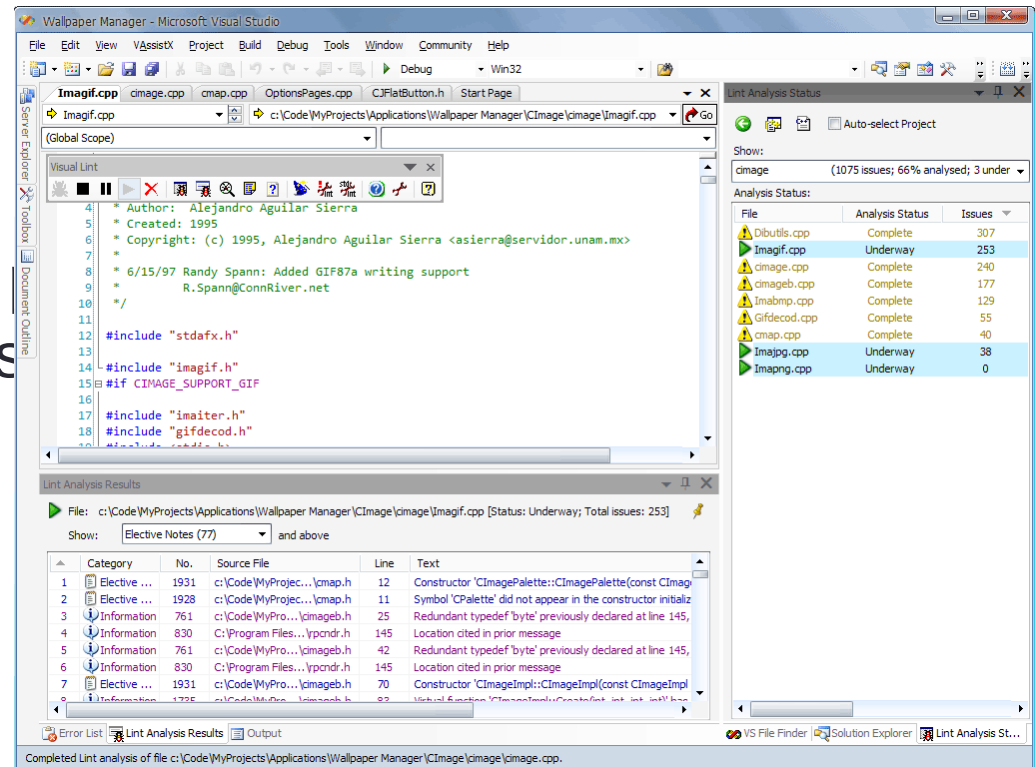
# IDEs

- Existem várias IDEs de mercado, algumas delas
  - Dev c++
  - **Code::Blocks**
  - Microsoft Visual Studio
  - Netbeans
  - Notepad++
- O importante é notar que independente da plataforma utilizada o código limpo, indentado e comentado é extremamente recomendável



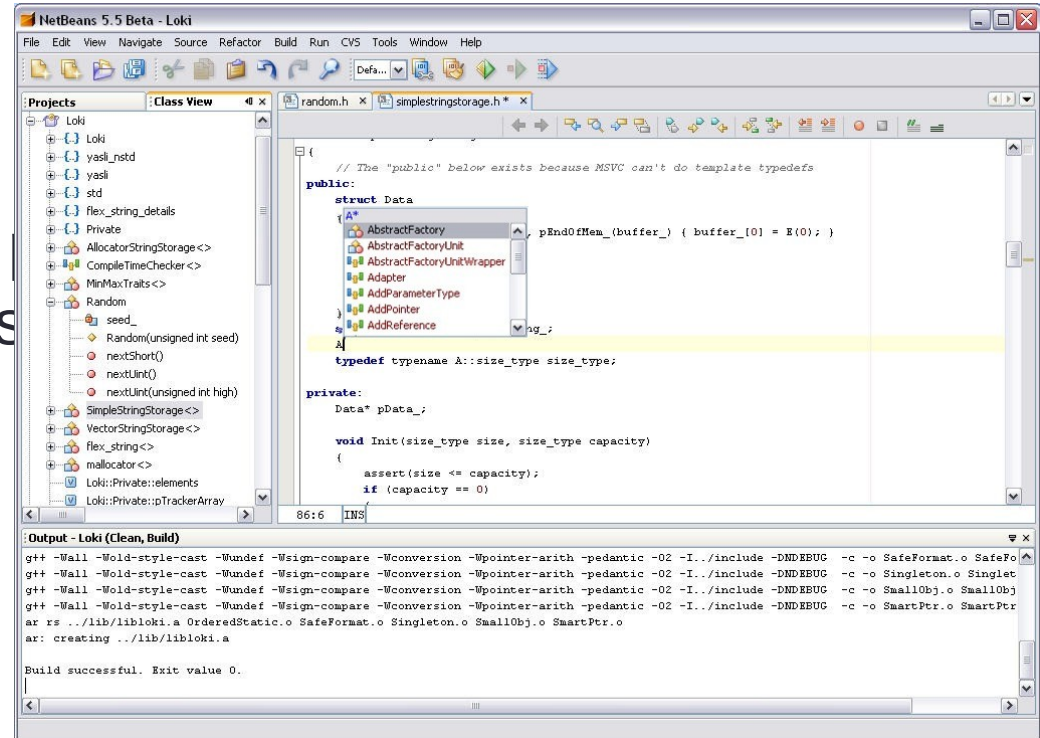
# IDEs

- Existem várias IDEs de mercado, algumas delas
  - Dev c++
  - Code::Blocks
  - Microsoft Visual Studio**
  - Netbeans
  - Notepad++
- O importante é notar que independente da plataforma utilizada o código limpo, indentado e comentado é extremamente recomendável



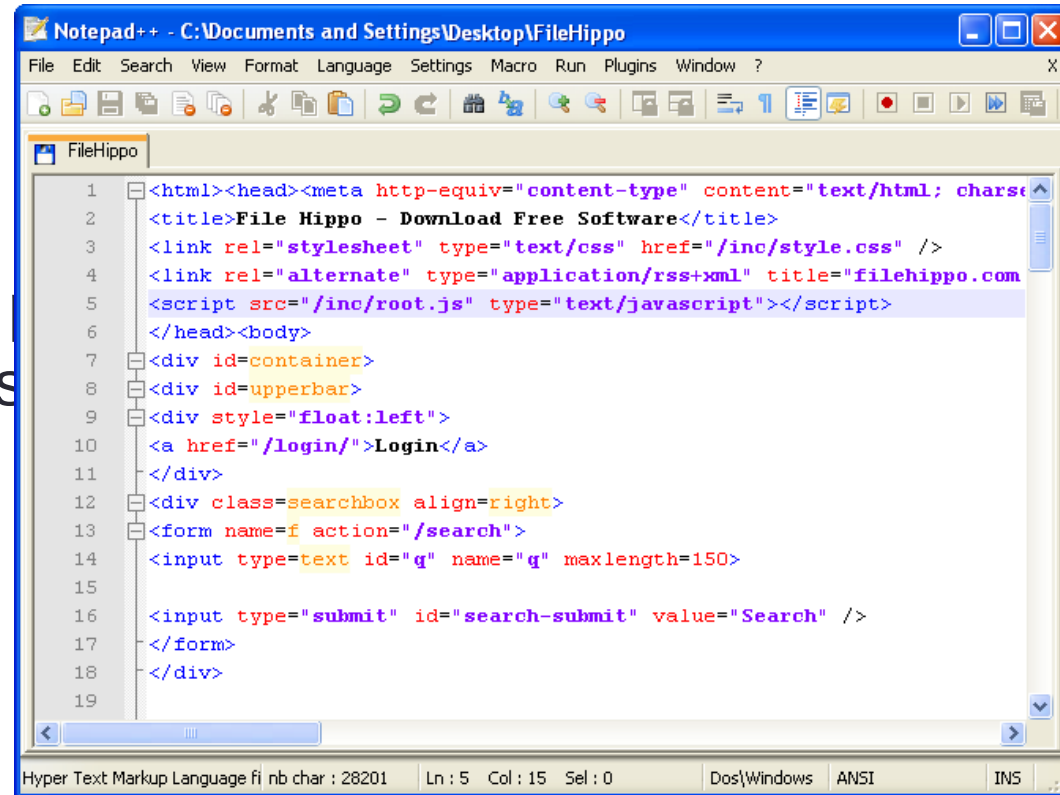
# IDEs

- Existem várias IDEs de mercado, algumas delas
  - Dev c++
  - Code::Blocks
  - Microsoft Visual Studio
  - **Netbeans**
  - Notepad++
- O importante é notar que independente da plataforma utilizada o código limpo, indentado e comentado é extremamente recomendável



# IDEs

- Existem várias IDEs de mercado, algumas delas
  - Dev c++
  - Code::Blocks
  - Microsoft Visual Studio
  - Netbeans
  - **Notepad++**
- O importante é notar que, independente da plataforma utilizada, um código limpo, indentado e comentado é extremamente recomendável



The screenshot shows the Notepad++ application window with the file 'FileHippo' open. The code is HTML and is well-formatted with indentation and color-coding. The code includes a meta tag for content type, a title, a link to a stylesheet, an RSS link, and a script tag. The body contains a container div with an upper bar, a login link, and a search form with a text input and a submit button.

```
1 <html><head><meta http-equiv="content-type" content="text/html; charse
2 <title>File Hippo - Download Free Software</title>
3 <link rel="stylesheet" type="text/css" href="/inc/style.css" />
4 <link rel="alternate" type="application/rss+xml" title="filehippo.com
5 <script src="/inc/root.js" type="text/javascript"></script>
6 </head><body>
7 <div id=container>
8 <div id=upperbar>
9 <div style="float:left">
10 <a href="/login/">Login</a>
11 </div>
12 <div class=searchbox align=right>
13 <form name=f action="/search">
14 <input type=text id="q" name="q" maxlength=150>
15
16 <input type="submit" id="search-submit" value="Search" />
17 </form>
18 </div>
19
```



# IDEs

- Neste curso qualquer IDE poderá ser utilizada. A preferência é pela IDE Code::Blocks, onde provavelmente os trabalhos serão corrigidos
- Existe uma pequena mudança de nome ou local onde se encontra determinada função em cada uma das IDEs.
- Porém, as funcionalidades são as mesmas e uma rápida pesquisa na internet, a partir do momento que já se utiliza uma IDE, é suficiente para se adaptar a outra IDE.
- Os exemplos a seguir serão feitos utilizando a última versão do code::blocks 10.05

# Exemplo

- A seguir será feito um exemplo de compilação e execução de código utilizando inicialmente o prompt de comando e depois o code::blocks
- Dado um código inicial em C (que será explicado posteriormente)

```
#include<stdio.h>
```

```
int main(){  
    printf("Hello World!!!");  
}
```

# Compilação e Execução

- Para compilar um código existem basicamente duas maneiras após a configuração:
  - A IDE geralmente possui um botão que compila o código. A única condição é que o código esteja salvo
  - Para compilar por linhas de comando basta abrir o prompt de comando, ir até o diretório onde está o código e compilar.
    - Os comandos de compilação dependem de cada compilador e podem ser vistos no help
  - As IDEs permitem que a compilação do código e de projetos com vários arquivos seja feita de uma forma mais simples.
  - O compilador gera um executável (windows) ou um objeto (linux) no caminho onde ocorreu a compilação.

# Compilação e Execução

- Para executar o programa basta digitar o nome no prompt de comando (windows) ou ./nome (linux) e depois pressionar enter.
- Caso não haja nenhum tipo de comando e entrada/saída de dados o resultado não poderá ser avaliado.
- No caso da interface gráfica o programa é executado clicando-se duas vezes sobre ele.
- Caso não haja nenhum mecanismo que espere uma resposta do usuário após a execução, o programa é executado.
- Todavia, a tela aberta é fechada instantaneamente. Por isso é necessário criar UMA INTERFACE COM O USUÁRIO.

# Exemplo

- Salvar estas linhas de código em um arquivo chamado `first.c`
- Para compilar em linhas de comando, após navegar para o diretório do arquivo, digitar no *prompt* de comando:
  - `gcc first.c -o first.exe`
- O arquivo `first.exe` será criado e poderá ser executado digitando-se `first` em seguida *enter*

# Exemplo

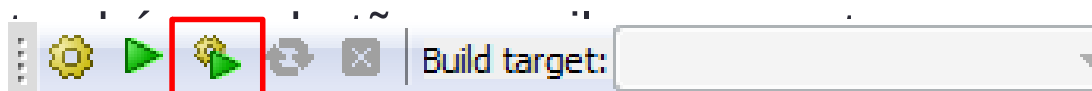
- Utilizando o code::blocks
  - *File -> New -> Empty File* ou *Arquivo -> Novo -> Arquivo em Branco*
  - Colar o código no arquivo.
  - Salvar o arquivo com *Ctrl+s* ou *Arquivo -> Salvar*
  - Clicar no botão compilar na barra que fica abaixo dos menus



- Depois clicar em executar



- Ou



# Exemplo

- O log da compilação deve aparecer na janela abaixo do código.
- Desta forma um código pode ser criado, alterado, compilado e modificado.
- Existe uma outra maneira de se criar um software utilizando projetos

# Projetos

- Algumas IDEs induzem a programação utilizando projetos independentemente da complexidade.
- Os projetos são interessantes para programas que possuem mais de um arquivo fonte como por exemplo os projetos que possuem arquivos *Header* e *Source*.
  - Nos arquivos *Header* são declaradas apenas as assinaturas da funções, ou seja, o nome, o tipo e as entradas
  - Nos arquivos *Source* são programadas as funções do Header e também outras que sejam necessárias. É necessário incluir o *Header* no arquivo *Source*.
- A compilação do projeto como um todo utilizando a IDE é mais fácil, pois não é necessário digitar o nome de todos os arquivos e todas as opções. Simplesmente o projeto é compilado da mesma forma que o arquivo.

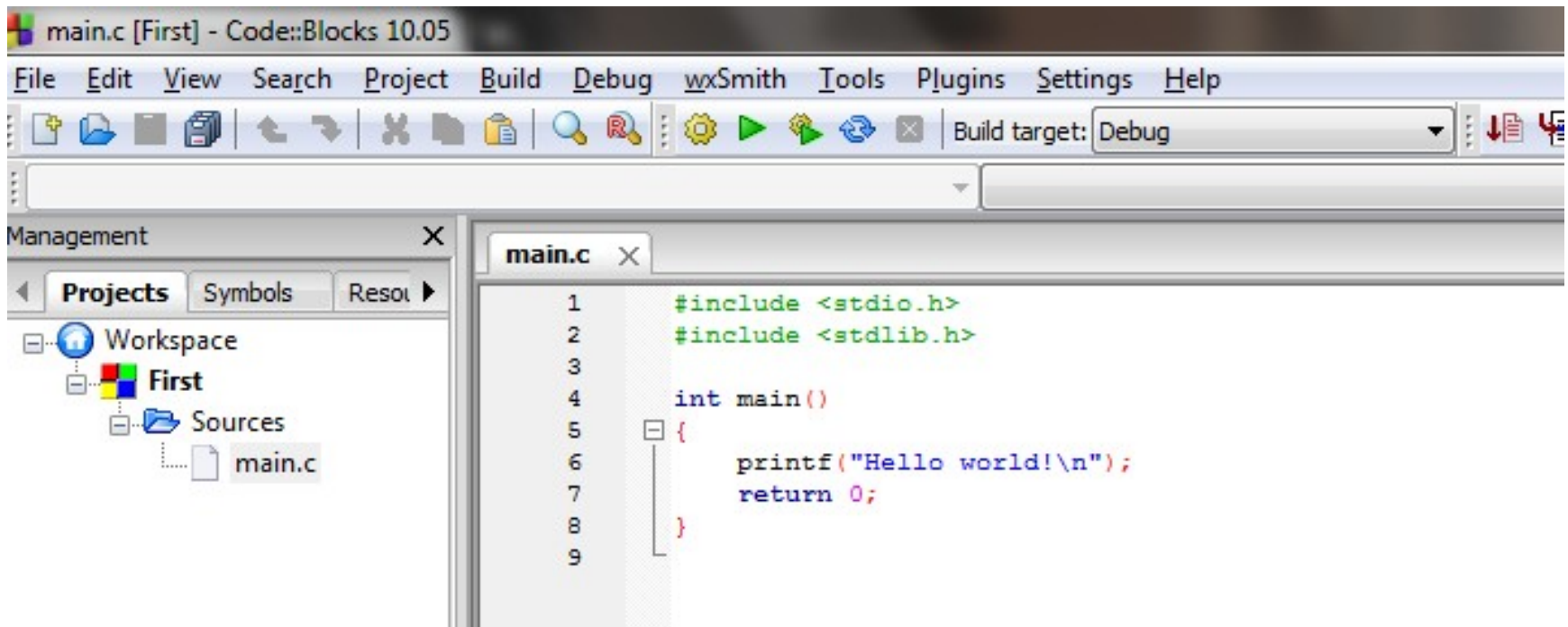


# Projetos

- Para iniciar um projeto
  - File -> New -> Project
  - Várias opções irão estar disponíveis. Neste momento a opção que nos interessa é Console Application
  - Depois um wizard é aberto e é só seguir as instruções:
    - Selecionar linguagem C
    - O nome do projeto, o diretório que ele irá ficar
    - As opções de compilação e debug que geralmente são padrão, a não ser que algum diretório necessite de mudança. A mudança nesta etapa não é comum

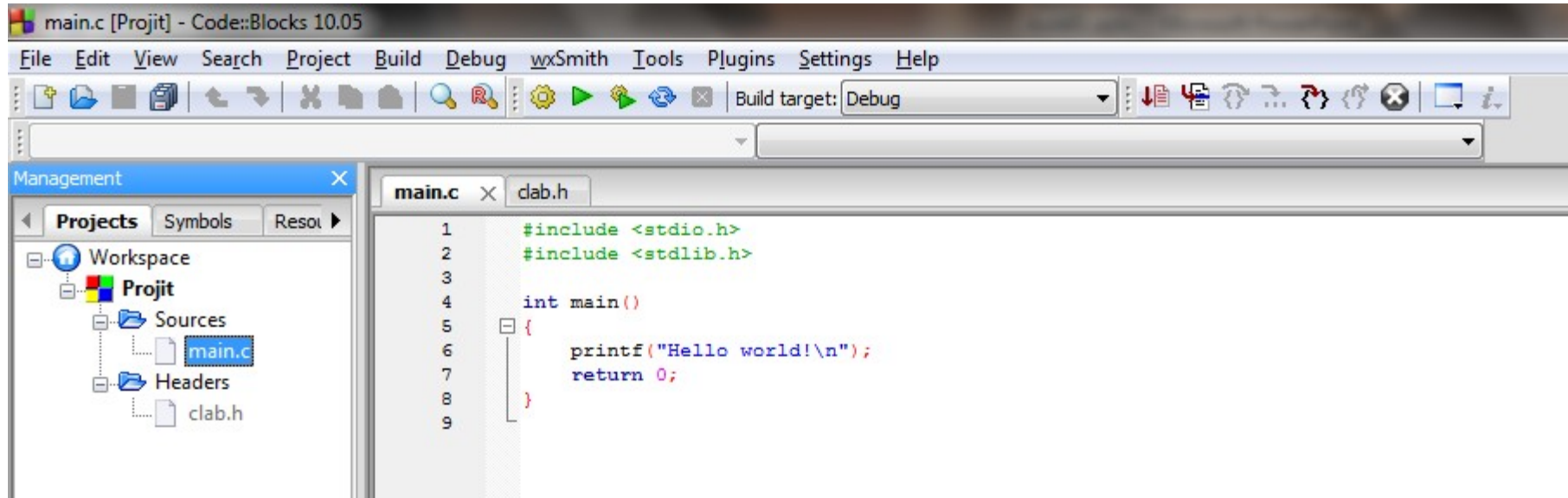
# Projetos

- O resultado é um projeto que possui a pasta source que contém um arquivo main.c (neste caso chamado de First).



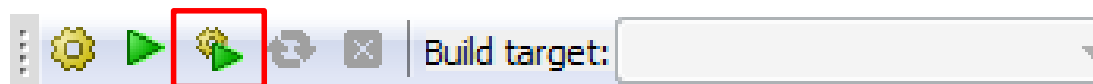
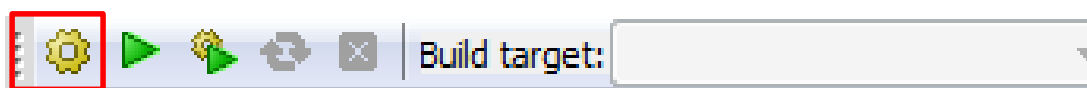
# Projetos

- Para criar a pasta Header ou outra pasta necessária adicione um arquivo Header ao projeto (nome.h). O projeto automaticamente cria a pasta.
- Resultado após adicionar ao projeto o arquivo clab.h

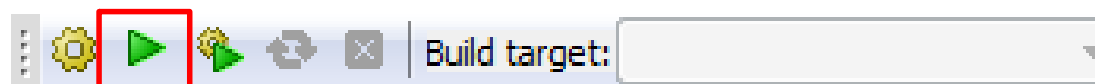


# Projetos

- Algumas IDEs já criam o projeto com a pasta *headers* mesmo que não exista nenhum arquivo *Header*
- Para compilar o projeto todo utiliza-se o mesmo botão citado anteriormente. Ele irá compilar todos os arquivos que foram incluídos no projeto



- O botão de execução irá executar apenas a função *main* do projeto.



# FIM

- Fazer os exercícios práticos da aula