University of Dhaka

Software Design Pattern Lab(CSE-3214)
Fall-2017

Suggestion Provider Text Editor

Md.Al-Helal(Roll-51)
Md.Inzamam-Ul-Haque(Roll- 85)
Md.Toukir Ahmed Sarker(Roll-125)

November 4, 2017

1 Introduction

A text Editor is used to edit plain text files. Text editors differ from word processors, such as Microsoft-Word or WordPerfect, in that they do not add additional formating information to documents. A plain text file is shown and edited by displaying all the characters the way they are involved in the file. ASCII is the most frequently used character set because of more frequent use of plain text files for programming and configuration and less frequent use of them for documentation (for instance, detailed instructions, user guides) as compared with the past.

But in our text editor we have added extra features such as auto completation of word and sentences,compiling of java and c files and error searching option from internet while an error occured during a c or java file compilation.

2 Motivation

Most of the text editors these days do not have the feature of auto completation of word and sentences. so users have to type the same thing over and over again, which is time consuming and not user friendly. To make the user experience better we have added this features. We have also provided the oportunity to compile code from our text editor because text editors are also used for coding and running them from terminal or command promt is boring and also time consuming because users have go back and forth to terminal and text editor.

3 Objectives

The main objective of our text editor is to make user experience better. It will save a lot of time of a user by providing suggestion for completing a sentence, by autocompletation of word.It will also work as the IDE of c, c++, java, python, LaTeX language because an user will be able to compile a c,c++,java,python,LaTex file from the text editor by clicking a compile button. We have also included searching option the solution of error which occur during compilation of a language. So users don't have to go to the browser and type the error himself and get suggestion from stackoverflow through data connection .

4 Project Features

1. Open files.
2. Save files.
3. Copy files.
4. Paste files.
5. Delete files.
6. Syntax provided for:
    - C
    - C++
    - Java
    - Assembly language
    - Python
    - LaTeX
7. Auto Completation of word.
8. Suggestion provided for completation of sentences.

9. User can compile language such as
    - C
    - C++
    - Java
    - Python
    - LaTeX
10. Suggestion provider when an error occured compiling a file
11. Browser Integration

5

6 Team Member Responsibilities

Md. Al-helal have implemented textpad features with strategy, factory method and Singlton patterns. Md. Inzamam have implemented suggestion features with adapter and comamnd patters. Md. Toukir Ahmed design the whole features and have contribution in command pattern and factory patters.

7 System Architecture

- JDK 8.0 or above
- Supported on Windows, Linux and MAC
- Minimum 2GB RAM required
- Minimum Memory 100MB

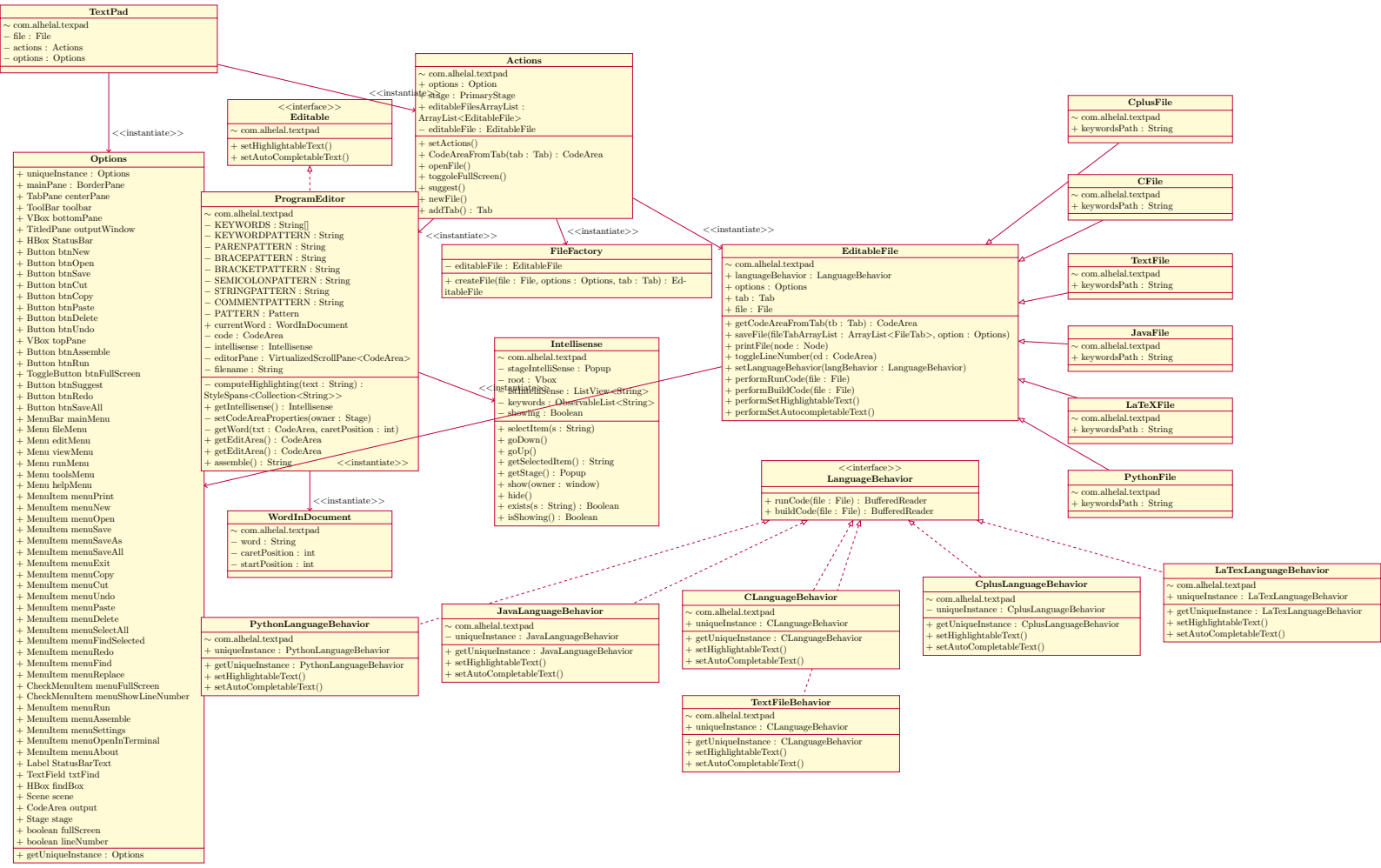
8 Platform, Library & Tools

We have used Java Platform to implement our text editor. We have used JavaFx library for developing our project.This project has all the frames prepared in Javafx.JavaFx offers stronger and manageable GUI components as compared to Swing and AWT. Modern Java GUI is represented by JAvaFx.We have also used FXMisc RichTextFX using maven dependency and Jsoup library.The tools we have used to implement the text editor is IntelliJ IDEA IDE.

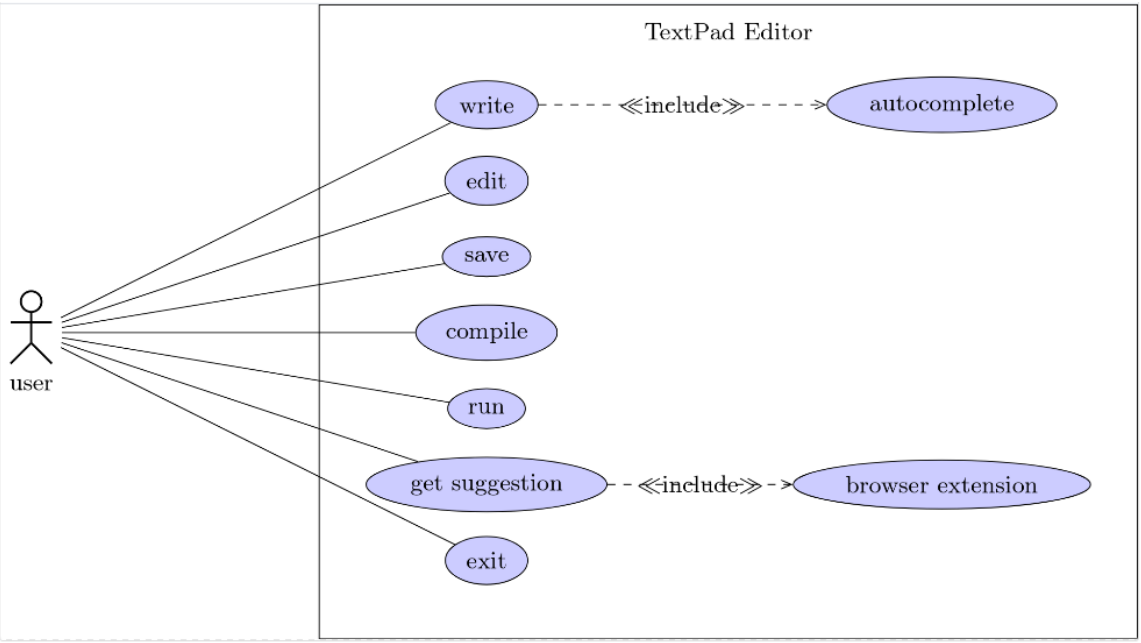
9 Limitations

- Data Connection is needed to get suggestion.
- Offline Suggestion are not added this version.
- Exact Suggestion limit is not fixed and also be improved in future.
- Autocomplete features are not fully implemented.
- Several features are not implemented in Textpad like font change,font size etc.
- Supports encoding=”UTF-8”.
- Response time is slow.

10 Class Diagram


TextPad
~ com.alhelal.texpad
− file : File
− actions : Actions
− options : Options
<<instantiate>>
<<instantiate>>
Actions
~ com.alhelal.textpad
+ options : Option
+ stage : PrimaryStage
+ editableFilesArrayList : ArrayList<EditableFile>
− editableFile : EditableFile
+ setActions()
+ CodeAreaFromTab(tab : Tab) : CodeArea
+ openFile()
+ toggoleFullScreen()
+ suggest()
+ newFile()
+ addTab() : Tab
<<interface>>
Editable
~ com.alhelal.textpad
+ setHighlightableText()
+ setAutoCompletableText()
Options
+ uniqueInstance : Options
+ mainPane : BorderPane
+ TabPane centerPane
+ ToolBar toolbar
+ VBox bottomPane
+ TitledPane outputWindow
+ HBox StatusBar
+ Button btnNew
+ Button btnOpen
+ Button btnSave
+ Button btnCut
+ Button btnCopy
+ Button btnPaste
+ Button btnDelete
+ Button btnUndo
+ VBox topPane
+ Button btnAssemble
+ Button btnRun
+ ToggleButton btnFullScreen
+ Button btnSuggest
+ Button btnRedo
+ Button btnSaveAll
+ MenuBar mainMenu
+ Menu fileMenu
+ Menu editMenu
+ Menu viewMenu
+ Menu runMenu
+ Menu toolsMenu
+ Menu helpMenu
+ MenuItem menuPrint
+ MenuItem menuNew
+ MenuItem menuOpen
+ MenuItem menuSave
+ MenuItem menuSaveAs
+ MenuItem menuSaveAll
+ MenuItem menuExit
+ MenuItem menuCopy
+ MenuItem menuCut
+ MenuItem menuUndo
+ MenuItem menuPaste
+ MenuItem menuDelete
+ MenuItem menuSelectAll
+ MenuItem menuFindSelected
+ MenuItem menuRedo
+ MenuItem menuFind
+ MenuItem menuReplace
+ CheckMenuItem menuFullScreen
+ CheckMenuItem menuShowLineNumber
+ MenuItem menuRun
+ MenuItem menuAssemble
+ MenuItem menuSettings
+ MenuItem menuOpenInTerminal
+ MenuItem menuAbout
+ Label StatusBarText
+ TextField txtFind
+ HBox findBox
+ Scene scene
+ CodeArea output
+ Stage stage
+ boolean fullScreen
+ boolean lineNumber
+ getUniqueInstance : Options
ProgramEditor
~ com.alhelal.textpad
− KEYWORDS : String[]
− KEYWORDPATTERN : String
− PARENPATTERN : String
− BRACEPATTERN : String
− BRACKETPATTERN : String
− SEMICOLONPATTERN : String
− STRINGPATTERN : String
− COMMENTPATTERN : String
− PATTERN : Pattern
+ currentWord : WordInDocument
− code : CodeArea
− intellisense : Intellisense
− editorPane : VirtualizedScrollPane<CodeArea>
− filename : String
− computeHighlighting(text : String) : StyleSpans<Collection<String>>
+ getIntellisense() : Intellisense
− setCodeAreaProperties(owner : Stage)
− getWord(txt : CodeArea, caretPosition : int)
+ getEditArea() : CodeArea
+ getEditArea() : CodeArea
+ assemble() : String
<<instantiate>>
<<instantiate>>
<<instantiate>>
<<instantiate>>
<<instantiate>>
FileFactory
− editableFile : EditableFile
+ createFile(file : File, options : Options, tab : Tab) : EditableFile
EditableFile
~ com.alhelal.textpad
+ languageBehavior : LanguageBehavior
+ options : Options
+ tab : Tab
+ file : File
+ getCodeAreaFromTab(tb : Tab) : CodeArea
+ saveFile(fileTabArrayList : ArrayList<FileTab>, option : Options)
+ printFile(node : Node)
+ toggleLineNumber(cd : CodeArea)
+ setLanguageBehavior(langBehavior : LanguageBehavior)
+ performRunCode(file : File)
+ performBuildCode(file : File)
+ performSetHighlightableText()
+ performSetAutocompletableText()
Intellisense
~ com.alhelal.textpad
− stageIntelliSense : Popup
− root : Vbox
− intellisense : ListView<String>
− keywords : ObservableList<String>
− showing : Boolean
+ selectItem(s : String)
+ goDown()
+ goUp()
+ getSelectedItem() : String
+ getStage() : Popup
+ show(owner : window)
+ hide()
+ exists(s : String) : Boolean
+ isShowing() : Boolean
<<instantiate>>
WordInDocument
~ com.alhelal.textpad
− word : String
− caretPosition : int
− startPosition : int
CplusFile
~ com.alhelal.textpad
+ keywordsPath : String
CFile
~ com.alhelal.textpad
+ keywordsPath : String
TextFile
~ com.alhelal.textpad
+ keywordsPath : String
JavaFile
~ com.alhelal.textpad
+ keywordsPath : String
LaTeXFile
~ com.alhelal.textpad
+ keywordsPath : String
PythonFile
~ com.alhelal.textpad
+ keywordsPath : String
<<interface>>
LanguageBehavior
+ runCode(file : File) : BufferedReader
+ buildCode(file : File) : BufferedReader
PythonLanguageBehavior
~ com.alhelal.textpad
+ uniqueInstance : PythonLanguageBehavior
+ getUniqueInstance : PythonLanguageBehavior
+ setHighlightableText()
+ setAutoCompletableText()
JavaLanguageBehavior
~ com.alhelal.textpad
− uniqueInstance : JavaLanguageBehavior
+ getUniqueInstance : JavaLanguageBehavior
+ setHighlightableText()
+ setAutoCompletableText()
CLanguageBehavior
~ com.alhelal.textpad
+ uniqueInstance : CLanguageBehavior
+ getUniqueInstance : CLanguageBehavior
+ setHighlightableText()
+ setAutoCompletableText()
TextFileBehavior
~ com.alhelal.textpad
+ uniqueInstance : CLanguageBehavior
+ getUniqueInstance : CLanguageBehavior
+ setHighlightableText()
+ setAutoCompletableText()
CplusLanguageBehavior
~ com.alhelal.textpad
− uniqueInstance : CplusLanguageBehavior
+ getUniqueInstance : CplusLanguageBehavior
+ setHighlightableText()
+ setAutoCompletableText()
LaTexLanguageBehavior
~ com.alhelal.textpad
+ uniqueInstance : LaTexLanguageBehavior
+ getUniqueInstance : LaTexLanguageBehavior
+ setHighlightableText()
+ setAutoCompletableText()

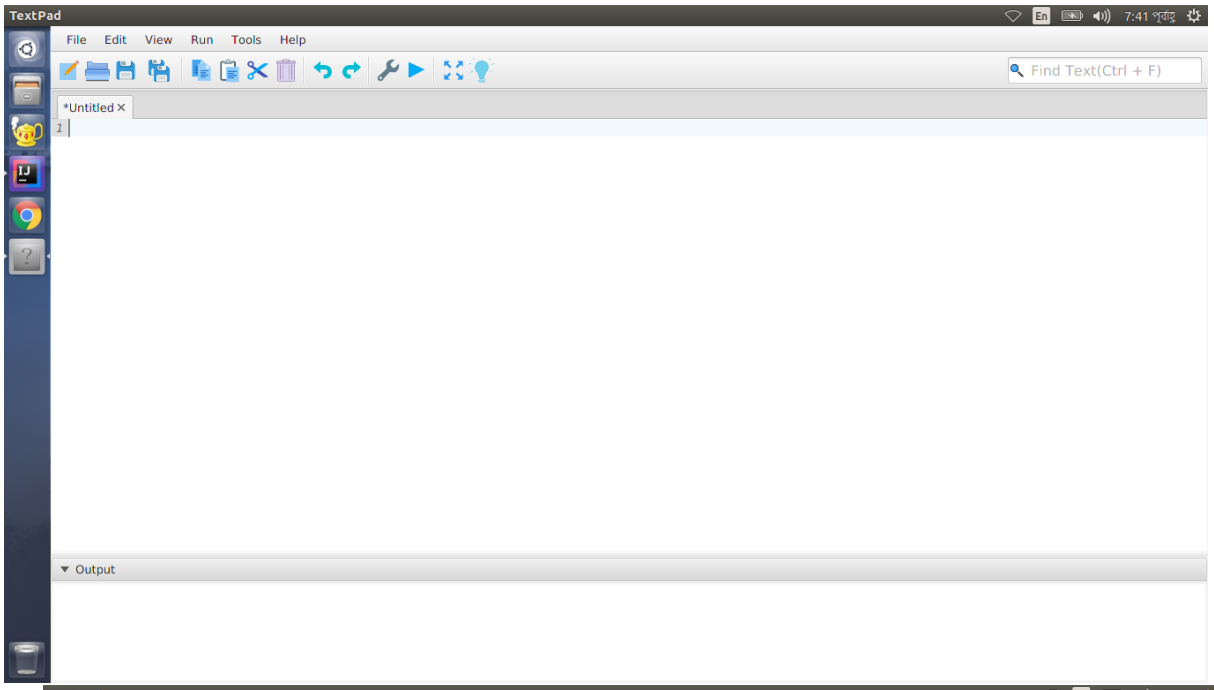
11 Use Cases

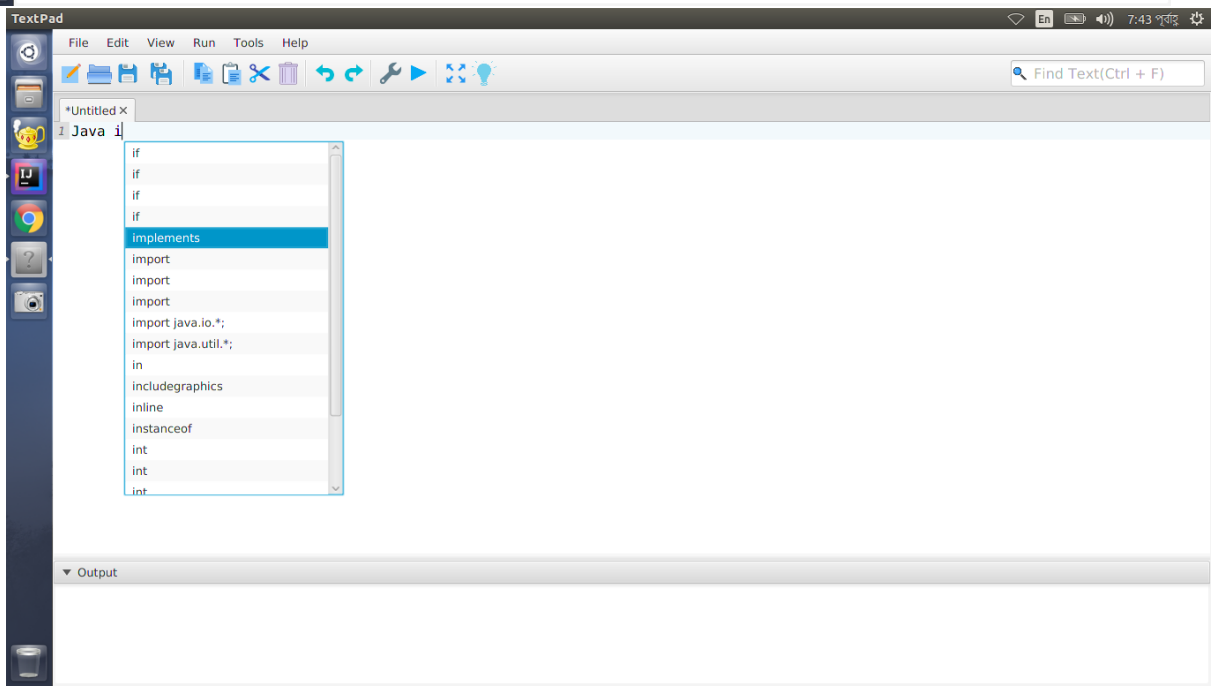


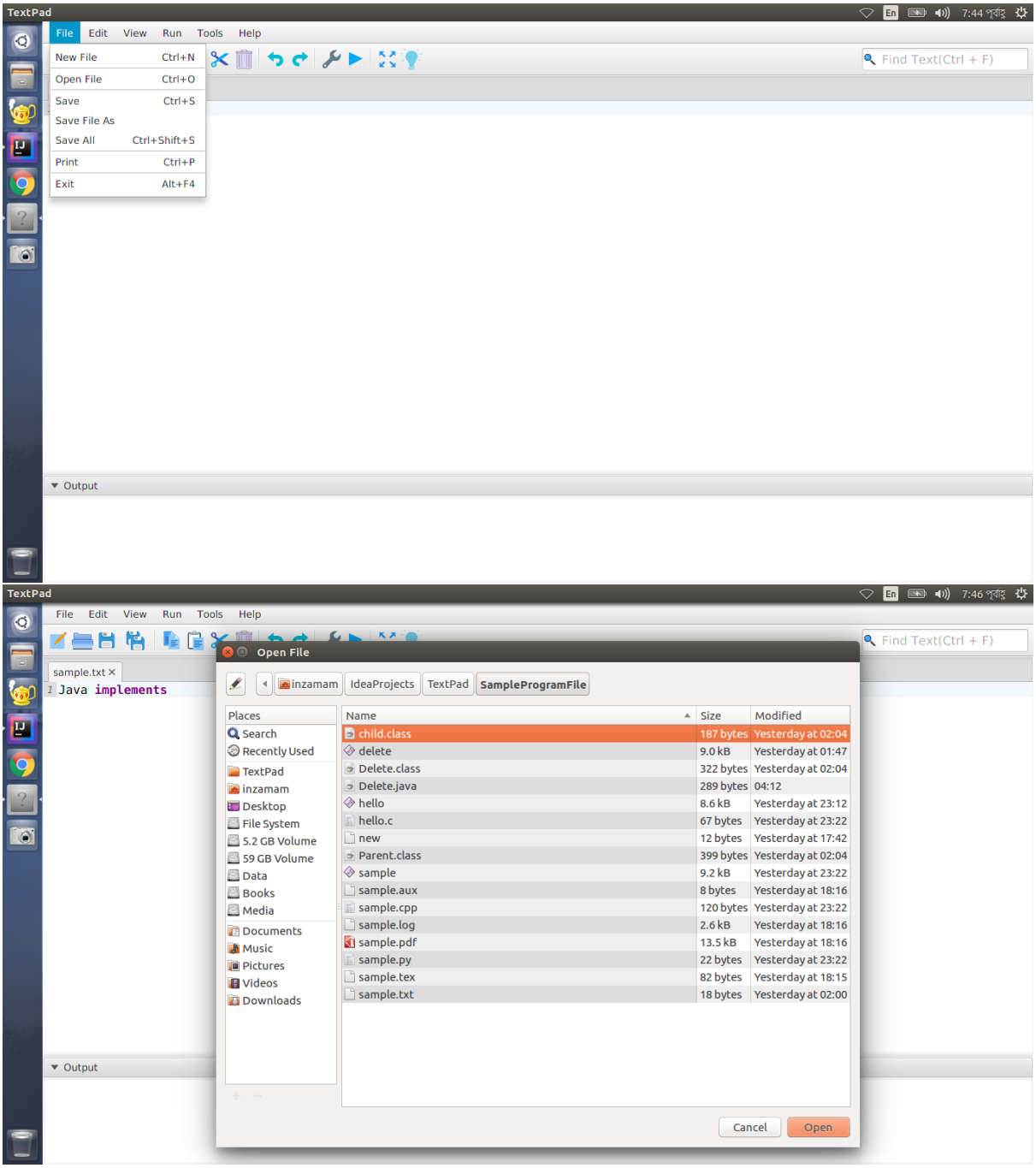
sectionPattern Used

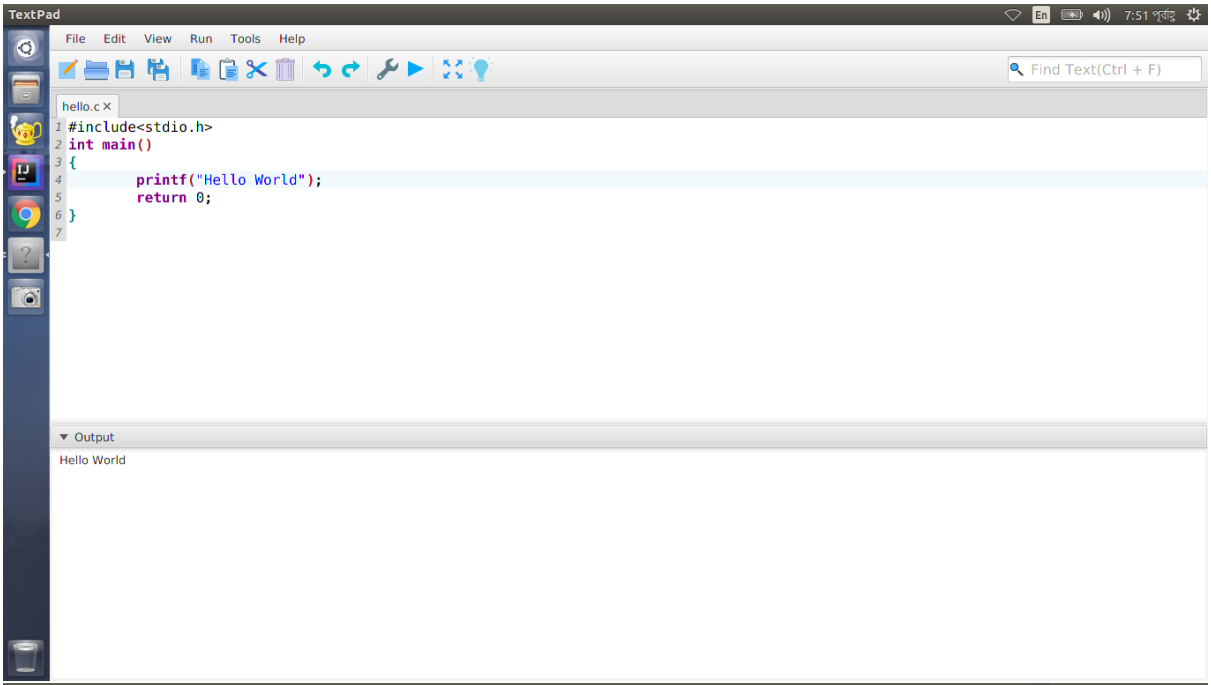
- Singleton
- Strategy
- Factory
- Adapter
- Command

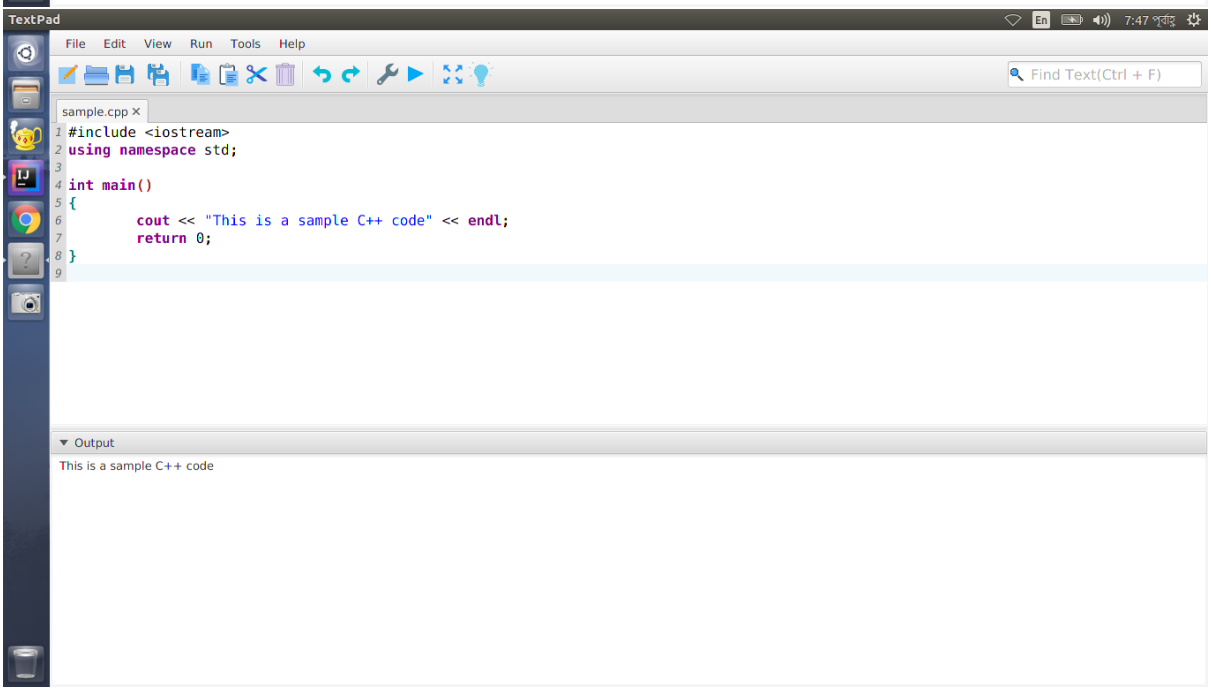
12 Screenshots

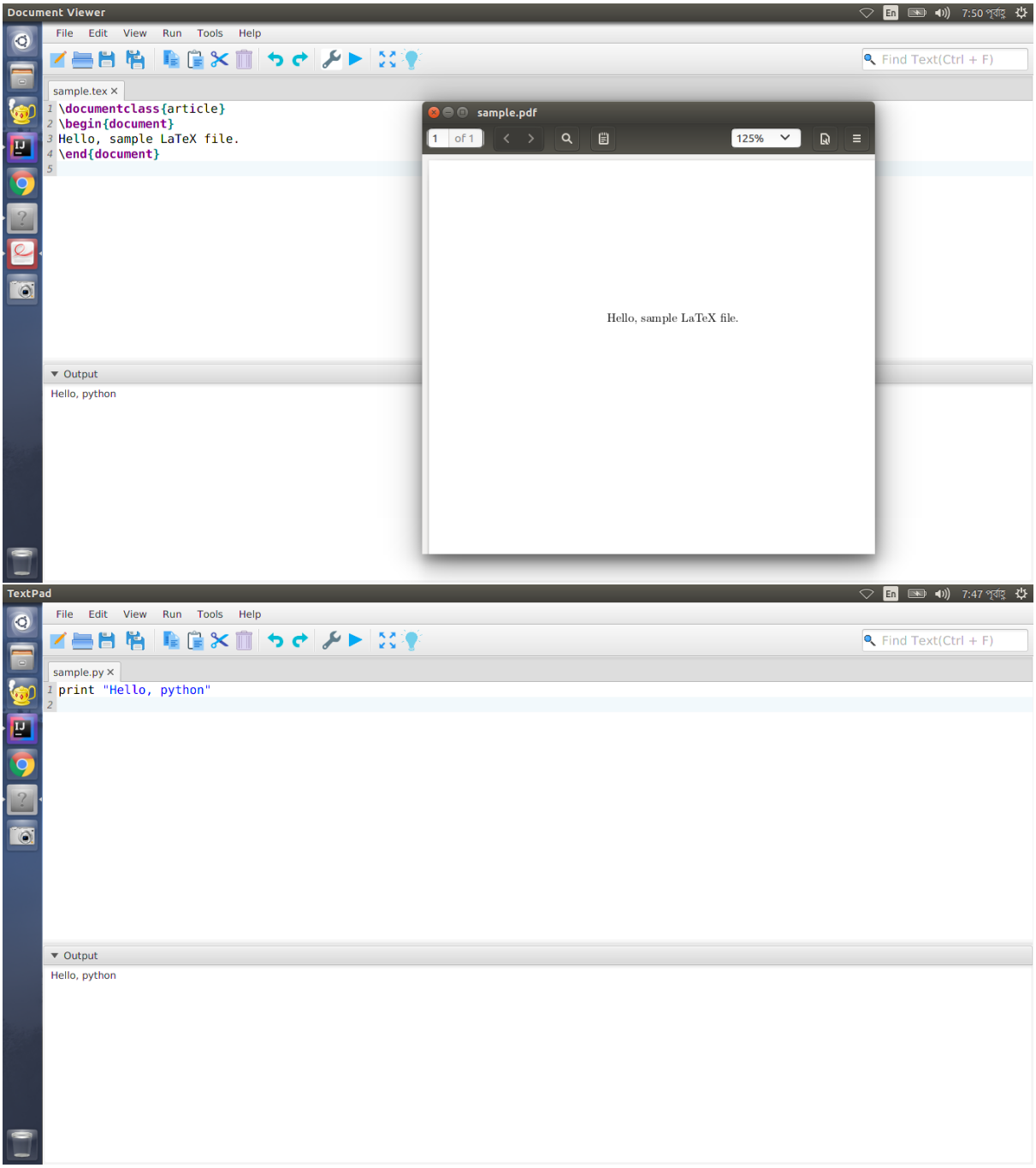


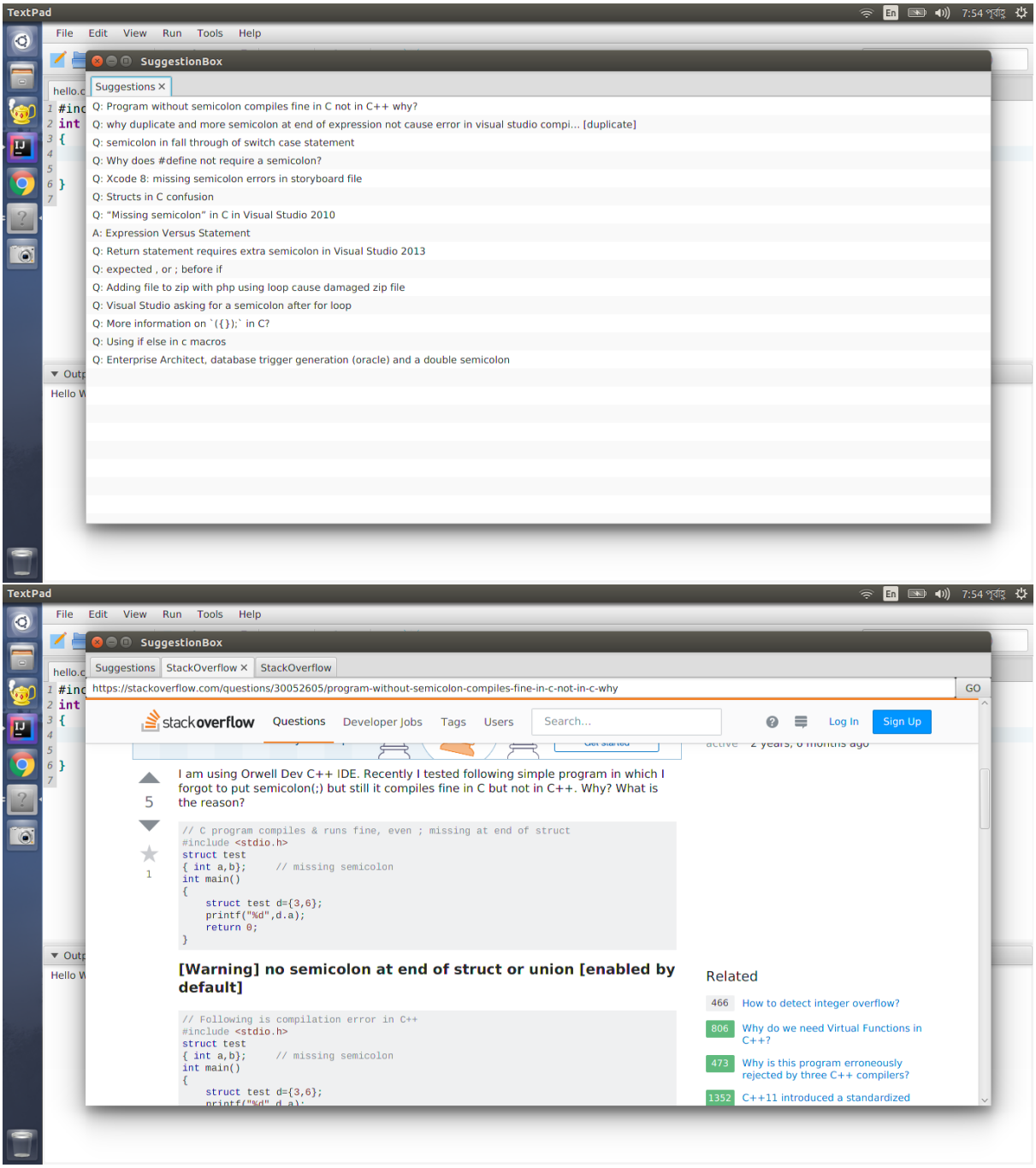


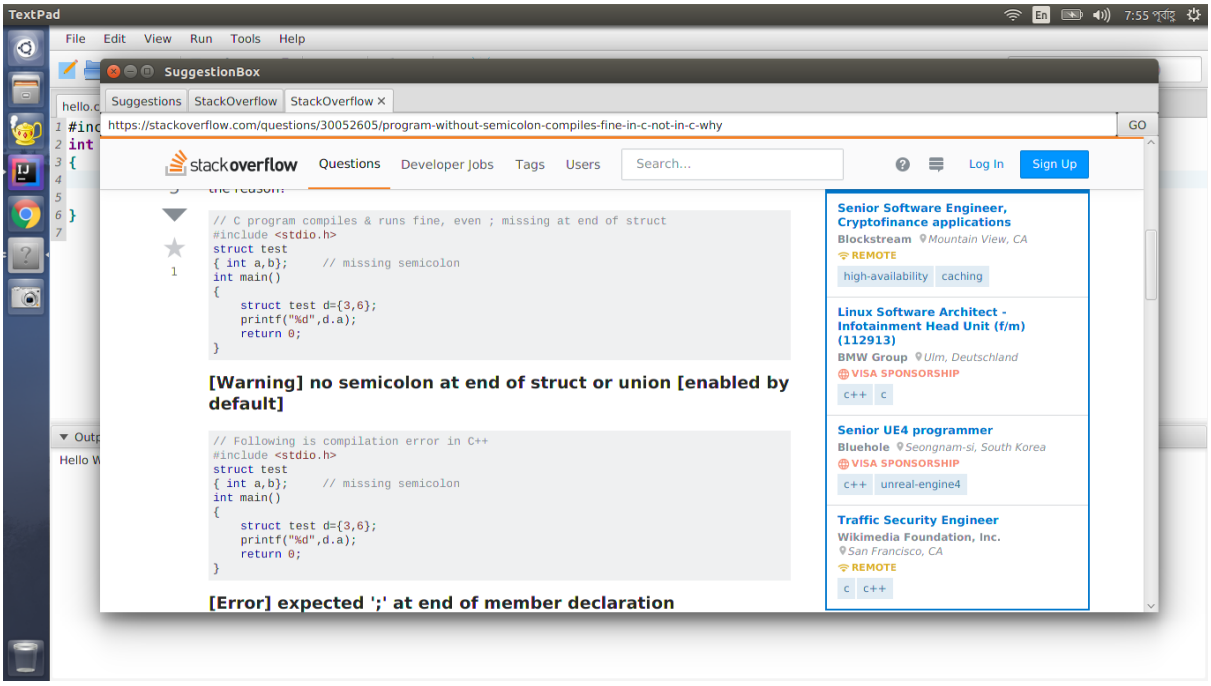

TextPad
File Edit View Run Tools Help
New File Ctrl+N
Open File Ctrl+O
Save Ctrl+S
Save File As
Save All Ctrl+Shift+S
Print Ctrl+P
Exit Alt+F4
Find Text(Ctrl + F)
Output
TextPad
File Edit View Run Tools Help
Find Text(Ctrl + F)
sample.txt
1 Java implements
Open File
inzamam IdeaProjects TextPad SampleProgramFile
Places
Search
Recently Used
TextPad
inzamam
Desktop
File System
5.2 GB Volume
59 GB Volume
Data
Books
Media
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
child.class 187 bytes Yesterday at 02:04
delete 9.0 kB Yesterday at 01:47
Delete.class 322 bytes Yesterday at 02:04
Delete.java 289 bytes 04:12
hello 8.6 kB Yesterday at 23:12
hello.c 67 bytes Yesterday at 23:22
new 12 bytes Yesterday at 17:42
Parent.class 399 bytes Yesterday at 02:04
sample 9.2 kB Yesterday at 23:22
sample.aux 8 bytes Yesterday at 18:16
sample.cpp 120 bytes Yesterday at 23:22
sample.log 2.6 kB Yesterday at 18:16
sample.pdf 13.5 kB Yesterday at 18:16
sample.py 22 bytes Yesterday at 23:22
sample.tex 82 bytes Yesterday at 18:15
sample.txt 18 bytes Yesterday at 02:00
Cancel
Open
Output


TextPad
File Edit View Run Tools Help
Find Text(Ctrl + F)
hello.c
#include<stdio.h>
int main()
{
printf("Hello World");
return 0;
}
Output
Hello World


TextPad
File Edit View Run Tools Help
Find Text(Ctrl + F)
sample.cpp
#include <iostream>
using namespace std;
int main()
{
cout << "This is a sample C++ code" << endl;
return 0;
}
Output
This is a sample C++ code


Document Viewer
File Edit View Run Tools Help
Find Text(Ctrl + F)
sample.tex
\documentclass{article}
\begin{document}
Hello, sample LaTeX file.
\end{document}
sample.pdf
1 of 1
125%
Hello, sample LaTeX file.
Output
Hello, python
TextPad
File Edit View Run Tools Help
Find Text(Ctrl + F)
sample.py
print "Hello, python"
Output
Hello, python


TextPad
File Edit View Run Tools Help
SuggestionBox
Suggestions ×
Q: Program without semicolon compiles fine in C not in C++ why?
Q: why duplicate and more semicolon at end of expression not cause error in visual studio compi... [duplicate]
Q: semicolon in fall through of switch case statement
Q: Why does #define not require a semicolon?
Q: Xcode 8: missing semicolon errors in storyboard file
Q: Structs in C confusion
Q: "Missing semicolon" in C in Visual Studio 2010
A: Expression Versus Statement
Q: Return statement requires extra semicolon in Visual Studio 2013
Q: expected , or ; before if
Q: Adding file to zip with php using loop cause damaged zip file
Q: Visual Studio asking for a semicolon after for loop
Q: More information on `({});` in C?
Q: Using if else in c macros
Q: Enterprise Architect, database trigger generation (oracle) and a double semicolon
TextPad
File Edit View Run Tools Help
SuggestionBox
Suggestions
StackOverflow ×
StackOverflow
https://stackoverflow.com/questions/30052605/program-without-semicolon-compiles-fine-in-c-not-in-c-why
GO
stackoverflow
Questions
Developer Jobs
Tags
Users
Search...
Log In
Sign Up
I am using Orwell Dev C++ IDE. Recently I tested following simple program in which I forgot to put semicolon(;) but still it compiles fine in C but not in C++. Why? What is the reason?
// C program compiles & runs fine, even ; missing at end of struct
#include <stdio.h>
struct test
{ int a,b};   // missing semicolon
int main()
{
struct test d={3,6};
printf("%d",d.a);
return 0;
}
[Warning] no semicolon at end of struct or union [enabled by default]
// Following is compilation error in C++
#include <stdio.h>
struct test
{ int a,b};   // missing semicolon
int main()
{
struct test d={3,6};
Related
466 How to detect integer overflow?
806 Why do we need Virtual Functions in C++?
473 Why is this program erroneously rejected by three C++ compilers?
1352 C++11 introduced a standardized



13 Conclusions

We have tried to implement various kind of design pattern through this project and learnt about various kind of design pattern. Finding various design pattern to implement our project was a bit difficult but we have successfully applied them in our project. In spite of tremendous hard work we had a lot of fun during this project. We have faced many difficulties to implement the project but we have overcome them successfully.

14 Future Plan

- Some features are not implemented we will complete them in future.
- We will convert our application into web app.
- Now our app gives only online error suggestion. We will make it offline.
- Language behavior will be improved in future.

References

- Head First Design Patterns by Bert Bates, Kathy Sierra, Eric Freeman, Elisabeth Robson
- http://fxmisc.github.io/richtext/javadoc/0.8.0/org/fxmisc/richtext/package-tree.html
- https://docs.oracle.com/javase/8/javafx
- https://www.tutorialspoint.com/designpattern/
- https://www.google.com
- https://www.stackoverflow.com
- https://www.youtube.com
- we would specialy thank our honourable course teachers for their continuous guidance and suggestions.