A **TextProcessor** is a software which helps to write texts on a file. A **ModernTextProcessor** is a **TextProcessor** which can highlight the syntaxes of a script, as well as offering the functionalities of a **TextProcessor.** A **Compiler** compiles a script/code and build executable files. A **ModernCompiler** has all the functionalities of a **Compiler** and **ModernTextProcessor.**

An **IDE** can build a project as well as offer all the functionalities of a **ModernCompiler.**

**Notepad** is a **TextProcessor. NotepadPlusPlus** is a **ModernTextProcessor. MinGW** is a **Compiler. Codeblocks & Netbeans** are both **IDE.**

While writing texts on a file, **Notepad** prints **“Notepad is writing text T on file F”,** where **T** is the text string and **F** is the file name string.

While writing texts on a file, **NotepadPlusPlus** prints **“Writing text T on file F using NotePad++”,** where **T** is the text string and **F** is the file name string. It highlights the syntaxes of a script printing “**Highlighting Keyword K in File F using Notepad++**”- where **K** is a string keyword and **F** is the file name string.

While compiling a code, **MinGW** prints “**Compiling X.cpp”** where X is the name of cpp file. It prints **“Building X.o and X.exe”** while building files, where **X** is the cpp file name.

All **IDE** shows same behaviour in some functionalities and differs in some. They are shown in the table. Think how they can be implemented using abstract class/inheritance.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | CodeBlocks | Netbeans | Parameters |
| Writing text on a file | **Writing on File F** | Same | **F: file name string** |
| Highlighting syntax | **Highlighting Keyword K** | Same | **K: keyword string** |
| Compiling a code | **Compiling X.cpp** | **Compiling X.java** | **X: file name string** |
| Building executables | **Building X.exe X.o** | **Building X.jar** | **X: file name string** |
| Build a project | **Building project P** | same | **P: project name string** |

**Your Task:**

Implement the whole scenario using inheritance/abstract class/interface. For the methods’ implementation, just use system.out.println. Remember to pass **X,F,K as String parameters in methods and Y as integers.**

Your first job is to draw a hierarchy diagram in a page. It will help you to understand the relation between the entities. Remember Khichuri and richfoods diagram? [5 Marks]

Then start from top and develop the classes one by one. [20 Marks]

Also, show a demo in the main function by instantiating an object of **Notepad, NotepadPlusPlus, MinGW, Codeblocks, Netbeans** individually and calling each one of their respective methods once. [5 marks]