**Competencies**

What is an imperative language?

imperative languages, in which the focus is

on how the computer should do it. \_

**What is a declarative language?**

declarative languages, in which the focus is on

what the computer is to do

**What is a translator?**

A **translator** is a computer program that performs the translation of a program written in a given [programming language](http://en.wikipedia.org/wiki/Programming_language) into a [functionally equivalent](http://en.wikipedia.org/wiki/Functionally_equivalent) program in a different computer language, without losing the functional or logical structure of the original code (the "essence" of each program).

What is a compiler?

A **compiler** is a [computer program](http://en.wikipedia.org/wiki/Computer_program) (or set of programs) that transforms [source code](http://en.wikipedia.org/wiki/Source_code) written in a [programming language](http://en.wikipedia.org/wiki/Programming_language) (the source language) into another computer language (the target language, often having a binary form known as [object code](http://en.wikipedia.org/wiki/Object_code)).[[1]](http://en.wikipedia.org/wiki/Compiler#cite_note-1) The most common reason for converting a source code is to create an [executable](http://en.wikipedia.org/wiki/Executable) program. Or complicated translator.

**What is an interpreter?**

 an ***interpreter*** is a [computer program](http://en.wikipedia.org/wiki/Computer_program) that directly [executes](http://en.wikipedia.org/wiki/Execution_(computers)), i.e. *performs*, instructions written in a [programming](http://en.wikipedia.org/wiki/Programming_language) or [scripting language](http://en.wikipedia.org/wiki/Scripting_language), without previously [compiling](http://en.wikipedia.org/wiki/Compiler) them into a [machine language](http://en.wikipedia.org/wiki/Machine_language) program. An interpreter generally uses one of the following strategies for program execution:

1. [parse](http://en.wikipedia.org/wiki/Parse) the [source code](http://en.wikipedia.org/wiki/Source_code) and perform its behavior directly
2. translate source code into some efficient [intermediate representation](http://en.wikipedia.org/wiki/Intermediate_representation) and immediately execute this
3. explicitly execute stored precompiled code[[1]](http://en.wikipedia.org/wiki/Interpreter_%28computing%29#cite_note-1) made by a [compiler](http://en.wikipedia.org/wiki/Compiler) which is part of the interpreter system

What is the Java Virtual machine?

Is a complicated Interpreter that translates code into java byte code.

What is the .NET framework?

**NET Framework** (pronounced *dot net*) is a [software framework](http://en.wikipedia.org/wiki/Software_framework) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) that runs primarily on [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows). It includes a large [class library](http://en.wikipedia.org/wiki/Class_library) known as [Framework Class Library](http://en.wikipedia.org/wiki/Framework_Class_Library) (FCL) and provides [language interoperability](http://en.wikipedia.org/wiki/Language_interoperability)(each language can use code written in other languages) across several [programming languages](http://en.wikipedia.org/wiki/Programming_language). Programs written for .NET Framework execute in a [software](http://en.wikipedia.org/wiki/Software) environment (as contrasted to [hardware](http://en.wikipedia.org/wiki/Computer_hardware) environment), known as [Common Language Runtime](http://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), an [application virtual machine](http://en.wikipedia.org/wiki/Process_virtual_machine) that provides services such as security, [memory management](http://en.wikipedia.org/wiki/Memory_management), and [exception handling](http://en.wikipedia.org/wiki/Exception_handling). FCL and CLR together constitute .NET Framework.

**What are the advantages of compiling to assembly code?**

The main advantage of execution via compilation is speed. Because the source program is translated directly to machine code, this program will most likely be faster than if it were interpreted.

**What are the advantages of interpreting a language?**

 In general, interpretation leads to greater flexibility and better diagnostics (error

messages) than does compilation. Because the source code is being executed

directly, the interpreter can include an excellent source-level debugger.

**Topics**

You should be able to discuss these topics and issues. You should also be familiar with the assigned Homework problems and study problems.

These are a comprehensive, but not complete, list of what you are expected to know.

***Section 1.1***

Why are there so many programming languages?

*Evolution., Special Purposes. Personal Preference, Expressive Power*

***Section 1.2***

Be able to explain these classifications.

* Imperative and declarative languages.

declarative languages, in which the focus is on

what the computer is to do, and the imperative languages, in which the focus is

on how the computer should do it. \_

Declarative languages are in some sense “higher level”; they are more in tune

with the programmer’s point of view, and less with the implementer’s point of view. Imperative languages predominate, however, mainly for performance reasons.

* Functional and logic languages.

employ a computational model based on the recursive definition of functions.

Logic programming systems allow the programmer to state a collection of *axioms*

from which theorems can be proven. The user of a logic program states a

theorem, or *goal*, and the language implementation attempts to find a collection

of axioms and inference steps (including choices of values for variables) that together

imply the goal. Of the several existing logic languages, Prolog is by far the

most widely used.

* Scripting languages.

are a subset of the von Neumann languages. They are distinguished

by their emphasis on “gluing together” components that were originally

developed as independent programs

* von Neumann languages.

the basic means of computation is the modification of variables.6

* Scripting languages.

are a subset of the von Neumann languages. They are distinguished

by their emphasis on “gluing together” components that were originally developed as independent programs

* Object-Oriented languages.

Picture it as interactions among semi-independent *objects*, each of which has both its own internal state and subroutines to manage that state

***Section 1.3***

Why study programming languages?

***Section 1.4***

Be able to explain these implementation methods.

* Translator.
* Compiler.
* Interpreter.
* Virtual machine.
* Just-in-time compilers.

In some cases a programming system may deliberately delay compilation until

Dynamic and just-in-time Compilation the last possible moment

* Microcode.

On some machines (particularly those designed before the mid-1980s), the

Microcode (firmware) assembly-level instruction set is not actually implemented in hardware but in

fact runs on an interpreter. The interpreter is written in low-level instructions

called *microcode* (or *firmware*), which is stored in read-only memory and executed by the hardware

***Section 1.5***

Be able to explain these programming tools.

* IDE.
  + Visual Studio.

**Microsoft Visual Studio** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](http://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](http://en.wikipedia.org/wiki/Computer_program) for [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), as well as [web sites](http://en.wikipedia.org/wiki/Web_site), [web applications](http://en.wikipedia.org/wiki/Web_application) and [web services](http://en.wikipedia.org/wiki/Web_service). Visual Studio uses Microsoft software development platforms such as [Windows API](http://en.wikipedia.org/wiki/Windows_API), [Windows Forms](http://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](http://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](http://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](http://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](http://en.wikipedia.org/wiki/Native_code) and [managed code](http://en.wikipedia.org/wiki/Managed_code).

* + Net Beans.

is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for developing primarily with [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), but also with other languages, in particular [PHP](http://en.wikipedia.org/wiki/PHP), [C](http://en.wikipedia.org/wiki/C_(programming_language))/[C++](http://en.wikipedia.org/wiki/C%2B%2B), and [HTML5](http://en.wikipedia.org/wiki/HTML5).[[3]](http://en.wikipedia.org/wiki/NetBeans#cite_note-3) It is also an [application platform](http://en.wikipedia.org/wiki/Platform_(computing)) framework for Java desktop applications and others.

* + Eclipse.
* In [computer programming](http://en.wikipedia.org/wiki/Computer_programming), **Eclipse** is an [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE). It contains a base [workspace](http://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Written mostly in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other [programming languages](http://en.wikipedia.org/wiki/Programming_language): [Ada](http://en.wikipedia.org/wiki/Ada_(programming_language)), [ABAP](http://en.wikipedia.org/wiki/ABAP), [C](http://en.wikipedia.org/wiki/C_(programming_language)), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [COBOL](http://en.wikipedia.org/wiki/COBOL), [Fortran](http://en.wikipedia.org/wiki/Fortran), [Haskell](http://en.wikipedia.org/wiki/Haskell_(programming_language)), [JavaScript](http://en.wikipedia.org/wiki/JavaScript), [Lasso](http://en.wikipedia.org/wiki/Lasso_(programming_language)), [Lua](http://en.wikipedia.org/wiki/Lua_(programming_language)), [Natural](http://en.wikipedia.org/wiki/NATURAL), [Perl](http://en.wikipedia.org/wiki/Perl), [PHP](http://en.wikipedia.org/wiki/PHP), [Prolog](http://en.wikipedia.org/wiki/Prolog), [Python](http://en.wikipedia.org/wiki/Python_(programming_language)), [R](http://en.wikipedia.org/wiki/R_(programming_language)),[Ruby](http://en.wikipedia.org/wiki/Ruby_(programming_language)) (including [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails) framework), [Scala](http://en.wikipedia.org/wiki/Scala_(programming_language)), [Clojure](http://en.wikipedia.org/wiki/Clojure" \o "Clojure), [Groovy](http://en.wikipedia.org/wiki/Groovy_(programming_language)), [Scheme](http://en.wikipedia.org/wiki/Scheme_(programming_language)), and [Erlang](http://en.wikipedia.org/wiki/Erlang_(programming_language)" \o "Erlang (programming language)). It can also be used to develop packages for the software [Mathematica](http://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.
* The initial [codebase](http://en.wikipedia.org/wiki/Codebase) originated from [IBM VisualAge](http://en.wikipedia.org/wiki/IBM_VisualAge).[[2]](http://en.wikipedia.org/wiki/Eclipse_%28software%29#cite_note-VisualAge-2) The Eclipse [software development kit](http://en.wikipedia.org/wiki/Software_development_kit) (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.
* Command line tools.
* Configuration Management.
* Source code control.
* Repositories.
* Documentation generator.

***Section 1.6***

Be able to explain the steps of the compilation process.  What are they? What is produced?