**Names:** Ryan Driscoll, Thom Lippincott, Jacob B. West

**Language:** F#

**Coding environment**: <version, OS, compiler used, **and URL if applicable**>

### **Distinct pieces of functionality:**

1. <functionality 1> : <associated code files>
2. <functionality 2> : <associated code files>
3. Etc.

### **Section 1: Comparison to C++, Java, Python, and/or Scala**

1. **What is the language’s philosophy?**

F# is a functional language managed by Microsoft. According to Microsoft’s documentation the language is data-oriented and the code commonly involves transforming data with functions (i.e. functional programming). Therefore when looking at other languages such as C++, Java, Python, and Scala, F# is most related to Scala due to the functional approach of solving problems.

1. **Compare and contrast your language in terms of the location it is used.**

Common places F# is used in industry is for machine learning and web development. The language has not gained the popularity such as Python for machine learning or javascript or C# for web development, however it is still used in these areas, specifically when data integrity is of concern.

1. **Compare and contrast your language in terms of where it excels and where it fails**
   * **Excels**

<answer here>

* + **Fails**

<answer here>

1. **Compare and contrast your language in terms of Portability, Simplicity, Orthogonality, AND Reliability.**

<Note: maximum of one page, and subheading are suggested>

* + **Portability**

F# can run in two separate environments. The first is within the .NET ecosystem and the second is within a browser, specifically through a few tools such as Binder and Fable, however these are smaller tools used more for prototyping and not full-scale production applications. Because of the limited areas in which F# can run, it is not as portable as other languages such as Java.

* + **Simplicity**

<answer here>

* + **Orthogonality**

<answer here>

* + **Reliability**

<answer here>

### **Section 2: Syntax, OOP**

1. **A) Write an example of one type of assignment expression in the language.   
   B) Then write the EBNF for a generic version of this assignment expression with all tokens defined.**

<answer here. Note: For simplicity, only consider numbers, strings, basic math, and/or function calls with no parameters. For example, a c++ assignment that needs an EBNF could be

int x = (2.5 + getValue()) / 3;>

1. **A) How does the language support extension etc. (single inheritance, interfaces, root object, class OOP, prototype OOP, other OOP, file importing, file extension, plugins, piping, module linking, etc.)?   
   <**answer here>

**B) Give an example.**

<answer here. Note: Unless it is esoteric, there will be some extensibility. If you say it does not support class-based OOP (the variety used almost exclusively in this class), so there is no extensibility, I will not believe you as that mean everything must be done in one file! Bash supports extensions! Max at 2 options.>

1. **A) How does the language handle module/namespace/packages/etc..**

F# has namespaces and modules in order to organize code into similar related areas based on functionality. Declaring a namespace occurs at the top of the file and every piece of code written after the declaration is considered to be in that namespace. Namespaces cannot have similar names when creating a new one. A function or value cannot be directly within a namespace, which the introduces the idea of modules.

Modules are considered to be the next level down in the organizational hierarchy. Modules will contain functions, values, types, etc. A module is a way to group code that has similar functionality and a namespace can have multiple modules within it. Within an F# program a developer can reference different namespaces and modules by using the open keyword, which is the equivalent to say an import keyword in C++ or Java. The referenced namespace or module must be in the same project or be included at the assembly level.

**B) What is the scope operator(s)? Alternatively, how to pick which variable if two code courses contain the same name?**

<answer here. Note: the scope operator in C++ is ::. In Java, it is . Python has a few options. >

**A) Does the language allow function overloading (name repetition), function redefinition, and/or function overriding?**<answer here>

**B) Give example syntax if it does.**

<answer here. Note: redefinition and overriding are NOT the same thing. >

### **Section 3: Binding, Type system, and data type range**

1. **Is the language static or dynamically typed? Give example syntax in code.**

In the Microsoft documentation F# is considered to be a static typed language. This might seem strange since the syntax implies otherwise, however behind the scenes at compilation the compiler will determine the type of every variable and if the compiler cannot determine type the developer must apply a type in the code in order to have successful compilation.

1. **Is the language static or dynamically scoped? Give an example in code.**

<answer here>

1. **How is the language read (in-fix, pre-fix, a mix, etc)? Give an example in code or in a diagram. (max of 3 examples).**

<answer here>

1. **What are the built-in data types and their ranges? (list 4-10, or send me a note if you believe that are less than 4)**

<answer here>

### **Section 4: Control flow, Function, specialties**

1. **What are the selection and repetition structures of the language, and what is their syntax?**

<answer here. >

1. **Are functions pass-by-value, pass-by reference, etc.? Give example syntax in code.**

<answer here>

1. **Describe at least two of the language specialties.**

Specialty one:

<answer here>  
Specialty two:  
 <answer here>