

**CS 320** ✕ **Programming Languages**  
**Research Project** ✕ **Spring 2022**  
Dr. Carolyn Pe Rosiene

*Please read this carefully.*

The goal of this project is to allow each student to fully investigate and study a particular programming language. You need to start early, as you will need to find materials from the library (or other university libraries) and other sources. Do not rely solely on the Internet. Textbooks and other references should be used as a main source, whenever possible.

Why are you given this assignment a month in advance? So you won't wait until two days before your presentation to do your research. Gathering your materials takes time; you will need to find suitable references such as getting an inter-library loan from the library (yes, real, paper books) which could take a week or two to come in. Start early!

### **Part 0 -- Language assignments**

Check out (verify) the language you are assigned on a Blackboard announcement. Study the language you are assigned in reference to the items below. Download a compiler or interpreter, or locate an online compiler/interpreter.

### **Part 1 -- Study your language**

Start by familiarizing yourself with your programming language. What are its strongest features, salient points, or weaknesses? What can it do that cannot be done in most other languages? Or just as easily? What makes it unique? What can it do really well (or not at all)? What are its highlights? Compile some sample programs and get used to its syntax.

### **Part 2 -- Select and design an algorithm**

You should start by picking an algorithm which exploits the features of the language. Design and debug the algorithm. The algorithm should be one that demonstrates the interesting features of the language and allows us to compare it with some of the other languages we have studied in class. Your algorithm should not be too simple, yet not too complicated. Focus on demonstrating the features of the language, not its length. For example, if your language is designed to manipulate vectors and matrices, design an algorithm that animates graphics and shows its transformations.

### **Part 3 -- Write and compile the program**

You should turn the algorithm into the target language. Write and debug the program. Install and use a compiler/interpreter for your program. Limit your sample program to two pages. Please use good programming style and comments. A "hello world" program (or similar) is not sufficient to demonstrate the features of a language.

### **Part 4 -- Give a presentation**

#### **General Outline:**

1. Introduction
  - What is this language design for?
2. History and Background
  - Where did this language arise from?
  - Who developed it and why?
3. Language Features – possibly multiple subsections here and show actual code snippets
  - What are its salient features?
  - What can this language do that others can't?
4. Advantages and Disadvantages
  - Why use this language?
  - Why not?
5. Applications
  - What kinds of applications are good for this language? Games, AI, graphics, web, medical analyses, etc.
6. Sample Program(s)
  - What is the language designed for?

- What applications fit the language best?
- What are the salient points of the language?

#### 7. Conclusions

- What's in store for this technology?
- Why might it succeed?
- Why might it fail?
- What should we remember/know about the language?

#### 8. References

You must use **at least three sources**. Sources may come from books, periodicals, academic/scholarly journals, manuals, consultation with industry professionals or organizations, or *reliable* web sites. The university library subscribes to on-line journals and more information may be found through <http://library.hartford.edu>. **The sources should not solely consist of web pages.** On-line books, periodicals, journals and scholarly papers do not count as web pages. Use MLA style citation, including [on-line sources](#).

You will give a short presentation about the language to the class. Each student will have some set amount of time for the entire presentation. Going overtime is counted against you but let me know if you think you will need extra time as it may be arranged. You are encouraged to use overhead slides, PowerPoint slides, web pages and/or short demonstrations during your presentation. A **demonstration using the compiler/interpreter** is strongly encouraged and would be **beneficial** to the class. Your presentation should be such that a student, after hearing your presentation, will be in a good position to understand simple code in that language. Practice your presentation ahead of time to ensure that you do not go over the allotted time. Ask a friend to listen and critique your talk. Please follow the outline provided.

When other students are presenting their material, please take notes; ask questions too. Presented material will be included in the final exam(or last test) in the form of "Matching" questions.

### Part 5 – Project Submission

Upload all your materials: presentation, code (zipped if multiple), reference books and articles (if e-copy), etc. Be sure to check the due date.

### Grading Rubric

Item	Points
Promptness – Start and end time; good use of your allotted time; not too short and not too long	5
Organization – Good layout; easy to follow	5
Clarity – Clear and concise	5
Visual Aids – Good use of visual aids including compiler/interpreter	10
Content – History and background	5
Content – Highlight salient features	10
Content – Advantage/disadvantage of the language	10
Content – Applications	5
Sample Program(s) – Good choice of program to show off language features	20
Conclusions – Future, wrap-up, etc.	5
References – At least 3 citations provided; use of reputable and credible sources; properly formatted	5
Q&A – Answers to questions are clear, accurate and complete with sufficient details to support the response	15