COMPILER DESIGN LAB SYLLABUS

1. Implementation of LEXR using LLVM.

2. Implementation of handwritten parser using LLVM

3. Generating code with the LLVM backend.

4. Defining a real programming language.

5. Write a recursive descent parser for the CFG language and implement it using LLVM.

6. Write a LR parser for the CFG language and implement it in the using LLVM.

7. Intro to Flex and Bison Modify the scanner and parser so that terminating a statement with "; b" instead of ";" results in the output being printed in binary.

8. Using LLVM-style RTTI for the AST and Generating IR from the AST.

9. Converting types from an AST description to LLVM types.

10. Emitting assembler text and object code.