Seat No.: _____ Enrolment No._____

# GUJARAT TECHNOLOGICAL UNIVERSITY

**M. E. - SEMESTER – II • EXAMINATION – SUMMER • 2013**

**Subject code: 1720103** **Date: 03-06-2013**

**Subject Name: Advance Compiler Design**

**Time: 10.30 am – 01.00 pm** **Total Marks: 70**

## Instructions:

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**

| | | | |
|---|---|---|---|
| **Q.1** | **(a)** | What is compiler? Explain the various phases of compiler in detail. Write down the output for the following expression after each phase **a=b*c-d+e** | **07** |
| | **(b)** | Define:<br>Linker, Loader, Token, Symbol table, Sentinel, Pattern | **07** |
| **Q.2** | **(a)** | What is ambiguity? How you eliminate ambiguity from dangling-else grammar? | **07** |
| | **(b)** | Construct a syntax-directed translation scheme that translates arithmetic expression from postfix notation into infix notation. Give annotated parse tree for the inputs 95-2* and 952*- | **07** |
| | | **OR** | |
| | **(b)** | Construct a syntax-directed translation scheme that translates integers into roman numerals. | **07** |
| **Q.3** | **(a)** | Define:<br>DFA, NFA, Prefix of string, Handle pruning, Viable prefix, Predictive parsing, Left recursion | **07** |
| | **(b)** | What is left-factoring? How you eliminate it? | **07** |
| | | **OR** | |
| **Q.3** | **(a)** | What is top-down parsing? Explain nonrecursive predictive parsing with example. | **07** |
| | **(b)** | What is operator-precedence parsing? Explain it. | **07** |
| **Q.4** | **(a)** | Explain in brief various error-recovery strategies used by parser. | **07** |
| | **(b)** | Write a short note: LR parsers | **07** |
| | | **OR** | |
| **Q.4** | **(a)** | Explain specification of a simple type checker with example. | **07** |
| | **(b)** | What is polymorphic function? Explain it. | **07** |
| **Q.5** | **(a)** | Explain activation record in brief. | **07** |
| | **(b)** | Write a short note: three-address code | **07** |
| | | **OR** | |
| **Q.5** | **(a)** | Explain various code-improving transformations with example. | **07** |
| | **(b)** | Write a short note: global data flow analysis | **07** |

**********