MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL 576104
(Constituent College of Manipal University)

FIFTH SEMESTER B.Tech. (IT) DEGREE END SEMESTER EXAMINATION DECEMBER– 2015
SUBJECT: SYSTEMS PROGRAMMING – ICT 307
(REVISED CREDIT SYSTEM)

TIME: 3 HOURS   **02/12/2015**   **MAX. MARKS: 50**

Instructions to candidates
- Answer any **FIVE FULL** questions.
- Missing data, if any, may be suitably assumed.

1A. Consider the following code and do the following:
    i.   Write the three address code
    ii.   Represent in terms of basic blocks
    iii.   Perform the necessary optimizations.

```
w=0;
x=x+y;
if(x>z)
  {y=x;
   x++; }
else
  { y=z;
   z++; }
w=x/z;
```

1B. Differentiate between L-attributed and S-attributed definitions. Give an example for each.

1C. Find a regular expression corresponding to the language of all strings over the alphabet { a, b } for the following.
    i.   Strings that contain exactly two a's.
    ii.   Strings that contain not more than one occurrence of the string "aa"   (5+3+2)

2A. Construct SLR parsing table for the following grammar also parse the string : {a{b a}}

```
T → B | { L }
L → T L | B
B → a | b
```

2B. Using the properties of LL(1) grammars check whether the following grammar is LL(1) or not ? (Not by constructing the table)

```
S → i C t S E  |  a
E → eS | ε
C → f
```

2C. What is conditional expansion during macro processing? Explain.   (5+3+2)

3A. Explain static and stack allocation of procedures with an example for each by considering three procedures (one main procedure) and two procedure calls from the main procedure.

3B. Write grammar for the language $L = \{a^n b^m : n \neq 2m\}$.

3C. What does parsing mean in the context of compiler design?   (5+3+2)

4A. Construct the CLR parsing table for the following grammar and parse the given string.

```
S → AA
A → cA | d
```

String: ccdccd

4B. Eliminate the left recursion from the following grammar. Consider the order as S,E,T and F

```
S → E
E → E+T | T
T → E-T | F
F → E*F | id
```
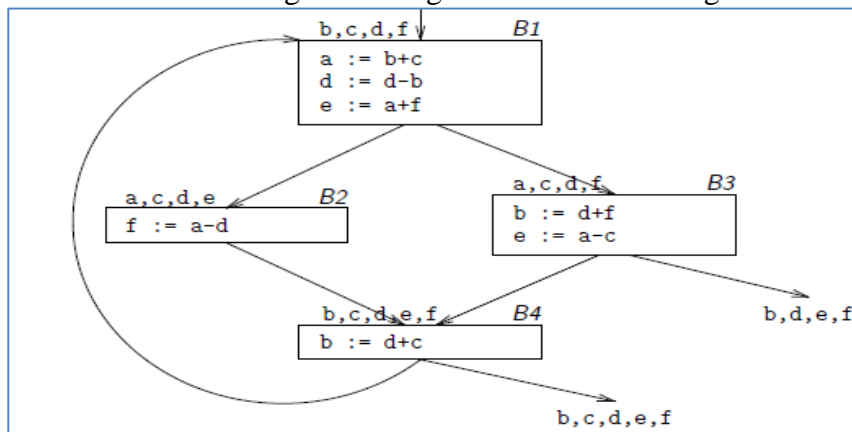
4C. Explain the datatypes used by the assembler.   (5+3+2)

5A. Consider the following grammar and do the following.
   i.   Construct the operator precedence table
   ii.   Construct the precedence functions table
   iii.   Show the movement of the parser for the input (id+id*id)

```
E→E+E | E*E | (E) |id
```

5B. Consider the following block diagram and find the usage count for all the variables.

5C. Obtain a DFA to accept odd number of 1's and even number of 0's $\sum=\{0,1\}$.   (5+3+2)

6A. Consider the following grammar for arithmetic expressions. Write the semantic actions to generate intermediate code in three address code notation. Also evaluate following expression using bottom up parsing by considering the semantic actions.

```
E→ E + T | T
T→ T * F | F
F→ (E) | id
Expression: 1+2*(2+4)
```

6B. What is Directed Acyclic Graph(DAG)? Construct DAG for the following expression.

```
a=b+c
t1=a*a
b=t1+a
c=t1*b
t2=c+b
a=t2+t2
```

6C. What are kernel and non-kernel items? Give an example.   (5+3+2)