Roll Number:

**Thapar Institute of Engineering and Technology, Patiala**
Computer Science and Engineering Department
**BE-MBA (VI Semester) EST**                                    **UCS802: Complier Construction**
*May 2022*
Time: *2 Hours*; MM: *35*                                          Name of Faculty: *Shalini Batra*

Q1.   Consider the following syntax directed translation that computes the value of a string of
      **0**'s and **1**'s interpreted as a positive, binary integer.                                    (7)

$$B \rightarrow B_1 \mathbf{0} \ \{B.val = 2 * B_1.val\}$$

$$B \rightarrow B_1 \mathbf{1} \ \{B.val = 2 * B_1.val + 1\}$$

$$B \rightarrow \mathbf{1} \ \{B.val = 1\}$$

B is the only non-terminal in the SDT, **0** and **1** are two terminals in the SDT. *val* is the
synthesized attribute of B. Rewrite the SDT so that the underlying grammar is **not left
recursive**, and yet the same value is computed for the entire input string.

Q2    Consider the following expression:
         z = - (a+b)*(c+d) -(a+b+c)

   a)  Give the three-address code for the above expression.                                    (2)

   b)  Represent the above expression in the form of:                                           (5)

       i)      Syntax Tree    ii)      Directed Acyclic Graph      iii)      Quadruple

       iv)     Triples                  iv)     Indirect triples.

Q3a)  Differentiate between the following using suitable example:                               (3+2)

      i) Synthesized and Inherited attributes

      ii) Syntax tree and Parse tree

   b)  Explain Call by value with example.                                                      (2)

Q4a) What is activation tree? Explain various units of activation tree, in short, diagrammatically. (1+3)

b) Draw the activation tree for the code given below with x=9 and y=6. (3)

```
int x,y;
int gcd ( int u, int v)
{
        if (v == 0) return u;
        else return gcd (v, u%v);
}
void main()
{
        scanf("%d%d",&x,&y);
        printf("%d\n",gcd(x,y));
}
```

Q5. Consider the following C code segment.

```
for(i-0,i<n, i++)
{
        for (j=0; j<n;j++)
        {
                if(i%2)
                {
                        x += (4*j+5*i);
                        y +=(7 + 4*j);
                }
        }
}
```

a) Explain how the following code optimization techniques can be applied in the code given above:

(2 x 3)

i) Common Subexpression elimination

ii) Loop invariant movement

iii) Strength Reduction

b) Give the final code after applying all three code optimization techniques mentioned above. (1)