

Name :  
Reg No :

**{A}**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**  
**07 THRISSUR CLUSTER**

**SECOND SEMESTER M.TECH. DEGREE EXAMINATION APRIL 2018**

**Computer Science and Engineering**

**Computer Science and Engineering**

**07CS6102**

**ADVANCED COMPILER DESIGN**

**Time : 3 hours**

**Max.Marks: 60**

Answer all six questions. Part 'a' of each question is compulsory.

Answer either part 'b' or part 'c' of each question

| <b>Q.no.</b>         | <b>Module 1</b>  | <b>Marks</b> |
|----------------------|--|--------------|
| <b>1a</b>            | Describe about L-Attributed definitions with an example.   | <b>4</b>     |
| <b>b</b>             | How syntax directed translation is applied to implement array data types?  | <b>5</b>     |
| <b>c</b>             | Write the procedure for implementing while statements using LL parsing.  | <b>5</b>     |
| <b>Q.no.</b>         | <b>Module 2</b>  | <b>Marks</b> |
| <b>2a</b>            | How intermediate code will be generated using static single assignment form?<br>Write the static single assignment for the following expression<br>p=a+b<br>q=p-c<br>p=q*d<br>p=e-p<br>q=p+q | <b>4</b>     |
| <b>Answer b or c</b> |  |              |
| <b>b</b>             | Translate the arithmetic expression $a + -(b + c)$ into<br>i) A syntax tree ii) Quadruples iii) Triples iv) Indirect triples   | <b>5</b>     |
| <b>c</b>             | Illustrate about Back patching for Boolean expressions.  | <b>5</b>     |
| <b>Q.no.</b>         | <b>Module 3</b>  | <b>Marks</b> |
| <b>3a</b>            | Show the complete activation tree for the following program<br><br>int f ( int n)  | <b>4</b>     |

```

{
int t , s;
if(n < 2) return 1;

s=f(n-1); t=f(n-2); return s+t ; }

```

**Answer b or c**

- |          |  |          |
|----------|--|----------|
| <b>b</b> | Show the control stack implementation of quick sort program. | <b>5</b> |
| <b>c</b> | Write Baker's mark-and-sweep garbage collector algorithm.    | <b>5</b> |

| Q.no. | Module 4  | Marks |
|-------|---|-------|
| 4a    | Differentiate register allocation and register assignment with example. | 4     |

**Answer b or c**

- |          |   |          |
|----------|---|----------|
| <b>b</b> | Write an algorithm to partition three-address instruction into basic block and draw the flow graph for the following code fragment. | <b>5</b> |
|----------|---|----------|

```

for i from 1 to 10
  for j from 1 to 10
    a[i,j] =0;

```

- |          |   |          |
|----------|---|----------|
| <b>c</b> | Describe code generation algorithm with suitable example. | <b>5</b> |
|----------|---|----------|

| Q.no. | Module 5  | Marks |
|-------|---|-------|
| 5a    | How global common sub-expressions are eliminated. Explain with example. | 5     |

**Answer b or c**

- |          |  |          |
|----------|--|----------|
| <b>b</b> | Illustrate loops in flow graph.  | <b>7</b> |
| <b>c</b> | Describe about data-flow schema implementation using reaching definitions. | <b>7</b> |

| Q.no. | Module 6  | Marks |
|-------|---|-------|
| 6a    | Discuss about various constraints in code scheduling. | 5     |

**Answer b or c**

- |          |   |          |
|----------|---|----------|
| <b>b</b> | Write short note on basic-block scheduling.                     | <b>7</b> |
| <b>c</b> | Discuss about global code scheduling by illustrating algorithm. | <b>7</b> |