Roll No.												

PRESIDENCY UNIVERSITY
BENGALURU

SCHOOL OF ENGINEERING

TEST - 1

Even Semester: 2018-19 | **Date**: 05 March 2019

Course Code: CSE 217 | **Time**: 1 Hour

Course Name: Compiler Design | **Max Marks**: 40

Programme & Sem: B.Tech (CSE) & VI Sem | **Weightage**: 20%

Instructions:

(i) ***Answer all questions***

Part A

Answer **all** the Questions. **Each** question carries **five** marks. (2Qx5M=10)

1. List out four possible lexical errors in C language.
2. Choosing data structure to create symbol table will improve efficiency of compiler? Explain

Part B

Answer the Question. Question carries **ten** marks. (1Qx10M=10)

3. Given the grammar for Robot Operations

   Robot→walk Dir Robot
   Robot→pick Obj Robot
   Robot→drop Obj Robot
   Robot→stop
   Dir→north | south
   Obj→ box | ball

   Where the terminals are { walk, pick, drop, stop, north, south, east, west, box, ball}
   And Non-terminals are { Robot, Dir, Obj}

The corresponding parse table is given below

	walk	pick	drop	stop	north	south	box	ball
Robot	**Robot→walk Dir Robot**	**Robot→pick Obj Robot**	**Robot→drop Obj Robot**	**Robot→ stop**				
Dir					**Dir→ north**	**Dir→ south**		
Obj							**Obj→ box**	**Obj→ ball**

i. Derive the following string

walk south pick box drop ball stop

ii. Add a new rule or production to the grammar to include the directions east and west. Give the modified LL(1) parse table. Where east and west are another terminal.

Part C

Answer the Question. Question carries **twenty** marks. (1Qx20M=20)

4. Given the grammar

S→aABC | bCd

B→b

A→dC|daD|de

C→ cDf | ε

D→dD | ε

i: Check the grammar is deterministic or not? If not perform Left factoring and make the grammar suitable for top down parsing
ii: Compute the first set
iii: Compute the follow set
iv: Construct the parse table. Justify why it is LL(1) grammar
v: Derive the string : adadbcf

Roll No.												

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY
BENGALURU

SCHOOL OF ENGINEERING

TEST - 2

Even Semester: 2018-19 | **Date**: 15 April 2019

Course Code: CSE 217 | **Time**: 1 Hour

Course Name: Compiler Design | **Max Marks**: 40

Program & Sem: B.Tech & VI Sem | **Weightage**: 20%

Instructions:

(i) ***Answer all questions***

Part A

Answer **both** the Questions. **Each** question carries **five** marks. (2Qx5M=10)

1. Having more than one IR and optimization is found in recent compilers. Why is it required? Explain with an example.
2. Given the Production

   $E \rightarrow E_1 + E_2$

   Write the semantic rules which satisfy the following requirements
   datatype are unsigned int, int, float and double,
   Type conversion is not allowed, Anything else is error.

Part B

Answer the Question. **The** Question carries **ten** marks. (1Qx10M=10)

3. Given the grammar for arithmetic operations

   E→ E+T/T
   T→ T*F/F
   F→ id/num/ (E)

   Construct the following for the input string ((a*b)+(a*b))+n
   a. Parse tree
   b. Abstract Syntax Tree(AST)
   c. Direct Acyclic Graph(DAG)
   d. Three address code

Part C

Answer the Question. The Question carries **twenty** marks. (1Qx20M=20)

4. Given the grammar

   E→E+T/T

   T→id+T/id

   a. Construct the PDA for the grammar
   b. Construct the LR(0) Parsing table
   c. Construct the SLR(1) Parsing table
   d. Check the string is valid or not: id+id-id

Roll No												

PRESIDENCY UNIVERSITY
BENGALURU

SCHOOL OF ENGINEERING

END TERM FINAL EXAMINATION

Even Semester: 2018-19 | **Date**: 23 May 2019

Course Code: CSE 217 | **Time**: 3 Hours

Course Name: Compiler Design | **Max Marks**: 80

Program & Sem: B.Tech & VI Sem | **Weightage**: 40%

Instructions:

(i) ***Answer All questions.***

Part A

Answer **all** the Questions. **Each** question carries **one** mark. (20Qx1M=20M)

1.

I. A bottom up parser generates
   a) Right most derivation
   b) Rightmost derivation in reverse
   c) Leftmost derivation
   d) Leftmost derivation in reverse

II. A grammar that produces more than one parse tree for some sentence is called
   a) Ambiguous
   b) Unambiguous
   c) Regular
   d) None of the mentioned

III. The optimization which avoids test at every iteration is called as_____
   a) Loop unrolling
   b) Loop jamming
   c) Constant folding
   d) None of the mentioned

IV. Shift reduce parsers are known as_____
   a) Top down Parser
   b) Bottom Up parser
   c) May be top down or bottom up
   d) None of the mentioned

V. Which of the following derivations does a top-down parser use while parsing an input string?
   a) Leftmost derivation
   b) Leftmost derivation in reverse
   c) Rightmost derivation
   d) Rightmost derivation in reverse

VI. Given the following expression grammar:
E -> E * F | F+E | F
F -> F-F | id
which of the following statement is true?
a) * has higher precedence than +
b) – has higher precedence than *
c) + and — have same precedence
d) + has higher precedence than *

VII. The output of a lexical analyzer is
a) A set of RE
b) Syntax Tree
c) Set of Tokens
d) String Character

VIII. Which of the following are Lexemes?
a) Identifiers
b) Constants
c) Keywords
d) All of the mentioned

IX. The graph that shows basic blocks and their successor relationship is called
a) DAG
b) Flow graph
c) control graph
d) Hamiltonion graph

X. Choose the correct FOLLOW for the Non terminal A for the following grammar
S→aABb
A→c/ ϵ
B→d/ ϵ
a) {$,b} b) {$,b,d} c) {$,d} d) {b,d}

XI. Choose the correct choice. For the given grammar
E→T+E
| T
T→i
a) Grammar is in LL(1)
b) Grammar is in LR(0)
c) Grammar is in SLR(1)
d) Grammar is in LL(1), LR(0) & SLR(1)

XII. Consider the grammar
S→A/a
A→a
Choose the correct choice
a) Grammar is in LR(0)
b) Grammar is not in LR(0) and having SR conflict
c) Grammar is not in LR(0) and having RR conflict
d) Grammar is not in LR(0) and having SR and RR conflict

XIII. Elimination of non-deterministic nature of grammar is termed as

a. Left recursion elimination b. Left factoring c. Ambiguity elimination

d. None of the above

XIV. Which of the statement is true which describes goals of code movement

i)To reduce the size of the code

ii)To reduce the frequency of the code

iii) To obtain time complexity

a. i and ii are true

b. i and iii are true

c. i , ii and iii are true

d. Only i is true

XV. Context Sensitive analysis deals with

a. Generation of syntax tree taking tokens

b. Performing type checking and type conversions

c. Analyze the context free grammar of the code

d. None of the above

XVI. Choose the correct FOLLOW for the Non terminal C for the following grammar

S→ABa / bCA

A→cBCD / ϵ

B→CdA / ad

C→eC / ϵ

D→bsf / a

a. {a,b,e,d,$}

b. {a,b,c,d,$}

c. {a,b,c,d)

d. {$,c}

XVII. Which one of the following is FALSE?

a. basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end.

b. Available expression analysis can be used for common subexpression elimination.

c. Live variable analysis can be used for dead code elimination.

d. x = 4 * 5 => x = 20 is an example of common subexpression elimination.

XVIII. 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);
    }
  }
}
```

Which one of the following is false?

a. The code contains loop invariant computation
b. There is scope of common sub-expression elimination in this code
c. There is scope of strength reduction in this code
d. There is scope of dead code elimination in this code

XIX. Consider the intermediate code given below:

```
1. i = 1
2. j = 1
3. t1 = 5 * i
4. t2 = t1 + j
5. t3 = 4 * t2
6. t4 = t3
7. a[t4] = –1
8. j = j + 1
9. if j <= 5 goto(3)
10. i = i + 1
11. if i < 5 goto(2)
```

The number of nodes and edges in the control-flow-graph constructed for the above code, respectively, are

a. 5 and 7
b. 6 and 7
c. 5 and 5
d. 7 and 8

XX. Consider the following source code :

```
c = a + b
d = c
c = c – e
a = d – e
b = b * e
b = d/b
```

Which of the following is the correct optimization of the given code?

a.
```
c = a + b
t = b * e
a = d – e
b = d/t
c = a
```

b.
```
c = a + b
d = c
c = c – e
a = d – e
b = d/b
```

c.
```
d = c
c = c – e
a = d – e
b = b * e
b = d/b
```

d. None of the above

Part B

Answer **all** the Questions. **Each** question carries **ten** marks. (4Qx10M=40M)

2. Consider the following code

```
i=1
sum=0
while(i<=20)
{
    sum=sum+a[i]*b[i]
    i=i+1
}
```

Assume the element size is 4 bytes

a. Generate the three address code for the given code.
b. Identify the leader in each block
c. Construct the control flow graph

3. Perform the code optimization for the following code

```
a=6
b=20
c=a-b
d=3*b-c
d=a
m=a-b
n=a-d
p=6-x
```

write the code after each operation in the order specified

a. Common Sub Expression Elimination
b. Copy Propagation
c. Dead Code Elimination
d. Constant Propagation

4. Given the code fragment res= a+a*(b-c)+(b-c)*d
   a. Generate the three address code for the code fragment
   b. Write the Quadruples representation
   c. Write the Triples representation
   d. Write the Indirect Triples representation
   e. Explain the advantages and disadvantages of each representation.

5. Given the set of instructions

```
d=1
x= (a+b)*d
y=a+b
z= y+ b++
p=a+b)
q=x+z+p
```

a. Perform the liveness analysis using use and def table
b. Construct the interference graph
c. Identify the minimum number of register required to generate code.
d. Generate the target code using x86 instruction set

Part C

Answer the Question. The question carries **twenty** marks. (1Qx20M=20M)

6. Given the grammar

L → W|D

W→ <while> (C) <begin> S <end>

D→<do> <begin> S <end> <while> (C)

C→ V <-V

V→ <id>|<num>

S→ AS| ε

A→ V-=V;A| ;

a. Construct the DFA for the given string
   while(a<=6)
   begin
   a-=1;
   end
b. Identify the tokens in the given string
c. Construct the predictive parsing table and check whether the given grammar is in LL(1) or not.
d. Parse the string which is given in (a)

Roll No												

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY
BENGALURU

SCHOOL OF ENGINEERING

SUMMER TERM / MAKE-UP END TERM EXAMINATION

Semester: Summer Term 2019 | **Date**: 25 July 2019

Course Code: CSE 217 | **Time**: 2 Hours

Course Name: Compiler Design | **Max Marks**: 80

Program & Sem: B.Tech (CSE) & VI Sem (2015 Batch) | **Weightage**: 40%

Instructions:

(i) ***Answer All questions***

Part A

Answer **all** the Questions. **Each** question carries **five** marks (4Qx5M=20M)

1. What are the drawbacks of one buffer scheme? How we will overcome from this drawbacks?

2. Construct a transition diagram for Unsigned number.

3. Consider the following input String

```
sum=0;
for(i=0;i<=5;i++)
{
    sum=sum+1;
}
```

Identify the tokens in the given string.

4. Find the FIRST and FOLLOW for the following grammar:

S→ABCDE
A→a | ε
B→b | ε
C→c
D→d | ε
E→e | ε

Part B

Answer **all** the Questions. **Each** question carries **ten** marks. (4Qx10M=40M)

5. Explain the various phases of compiler. Show the translation for an assignment statement a=b-c*80. Clearly indicate the output of each phase.

6. Consider the following code

```
i=10
sub=0
while(i>=0)
{
    sub=sub-a[i]+b[i]
    i=i-1
}
```

Assume element size is 4 bytes

a. Generate the three address code for given code
b. Identify the leader in each block
c. Construct the control flow graph

7. Given the code fragment res=a-a/(b+c)-(b+c)/d
   a. Construct AST and Three address code
   b. Construct DAG and Three address code

8. Consider the grammar given below

   E→5+T| 3-T

   T→VT'

   T'→*V | +V | ε

   V→a | b

   a. Compute FIRST and FOLLOW.
   b. Construct the predictive parsing table & check whether the given grammar is in LL (1) or not.

Part C

Answer the Question. **The** question carries **twenty** marks. (1Qx20M=20M)

9. Given the grammar

   S→(L) | a

   L→L,S | S

a. Construct the PDA for the grammar
b. Construct the LR(0) Parsing table and check the grammar is in LR(0) or not.
c. Construct the SLR(1) Parsing table and check the grammar is in SLR(1) or not.