International Islamic University Chittagong

Department of Computer Science and Engineering

B. Sc. in CSE

Final Exam, **Spring-2022**

Course Code: **CSE 3527** Course Title: **Compiler**

Time: 2 hours 30 minutes Full Marks: 50

(i) The figures in the right-hand margin indicate full marks

(ii) Course Outcomes and Bloom's Levels are mentioned in additional Columns

---

Part A

[Answer the questions from the followings]

1.	a)	Define LL(1) grammar and LR(k) parser. Why LR parsers are used in compiler design?	**CLO 2**	**R**	3
	b)	Construct CLR(1) and LALR(1) data flow diagram and parsing table for the following grammar: $S \rightarrow CC$ $C \rightarrow cC$ $C \rightarrow d$	**CLO 2**	U	7
2.	a)	Calculate the LR(0) Parsing for the given grammar- **E -> E + T | T** **T -> T * F | F** **F -> (E) | id**	**CLO 3**	App	**5**
2.	b)	What do you mean by Operator Precedence Grammar? With the help of following grammar parse the input string:- **"id + id * id "$** **T->T+T / T*T / id**	**CLO 2**	**E**	1+4 =5

Or,

2.	a)	For the following sub-problems, consider the following context-free grammar:	**CLO 3**	App	1+4 =5

$$S \rightarrow A\$ \quad (1)$$
$$A \rightarrow xAx \quad (2)$$
$$A \rightarrow C \quad (3)$$
$$B \rightarrow yBy \quad (4)$$
$$B \rightarrow C \quad (5)$$
$$C \rightarrow zBz \quad (6)$$
$$C \rightarrow wAw \quad (7)$$
$$C \rightarrow \lambda \quad (8)$$

i. What are the **terminals and non-terminals** of this grammar?

ii. Show the derivation of the string **xzzx$** starting from **S** (specify which production you used at each step), and give the **parse tree** according to that derivation.

2 b) Construct a **CLR parsing table** for the given context-free grammar — **CLO 2** — **E** — **5**

S-->AA

A-->aA | b

Part B

[Answer the questions from the followings]

3. a) Define the synthesized attributes and inherited attributes with example. Using syntax-directed definition draw the annotated parse tree and dependency graph for the sentence **(3*5+2)*2n.** — **CLO 1** — C — 5

S →En
E →E + T
E →T
T → T * F
T → F
F →(E)
F →digit

3. b) Write some examples of top-down parsing and bottom-up parsing. — **CLO 2** — E — 1+4 =5

Consider the following grammar-
E → E – E
E → E x E
E → id
Parse the input string id – id x id using a shift-reduce parser.

4. a) What is code optimization? — **CLO 3** — App — 1+4 =5

Consider the following code:
a = b * -c + b * -c
Write down the Quadruples of the given Code?

4. b) Write quadruple, triples and indirect triples for following expression : — **CLO 2** — U — 5

(x + y) * (y + z) + (x + y + z)

5. a) — **CLO 3** — **An** — 5

```
t1 = 4 * I
t2 = a [ t1]
t3=4 * I
t4 = b [ t3]
t5 = t2 * t4
t6 = prod + t5
t7 = I + 1
I = t7
If  I <= 20 goto 1
```

Write down the **DAG** to get the above information.

b) What do you know about the identifying the Basic blocks in target code generation? What do you know about the Peep hole optimization? | CLO 2 | An | 3+2 =5

Or,

5. a) Consider the following C codes and answer the questions (i) to (iii):- | CLO 3 | An | 5

```
int a[100],i,n=10,steps=0,temp;
      do{i=steps+1
            do {if(a[steps]>a[i]){
                  temp=a[steps];
                  a[steps]=a[i];
                  data[i]=temp;
                  i++;
            )while(steps<n);
            steps++;
}while(steps<n);
```

(i) Translate the C code into three-address code.
(ii) Identify the basic block in three-address code.
(iii) Construct the flow graph from the three-address code.

b) Draw the **syntax tree and DAG** for the expression: | CLO 2 | An | 5

a+a+a(a*a)+a+a+a*(b-c)+(b-c)*d