# COMSATS University Islamabad (Lahore Campus)

## ☐Midterm Exam ■Terminal Examination – Spring 2024

| | | | | | |
|---|---|---|---|---|---|
| Course Title: | Compiler Construction | Course Code: | CSC441 | Credit Hours: | 3(2,1) |
| Course Instructor/s: | M Bilal Arshad | Programme Name: | BSCS | | |
| Session: | SP21-BCS-A | Date: | 29-05-2024 | | |
| **Time Allowed:** | **2 hr 30 min** | **Maximum Marks:** | | **50** | |

**Important Instructions / Guidelines:**

- You are required to submit code files: (c / cpp )
- You are required to submit a zip folder consisting both of your files. Name your folder after your registration number. For example: FA20-RCS-022
- Submit your zip folder on google classroom.
- In case of cheating, **0** marks will be awarded

**Question No 1.** **Marks: 50**

***CLO: <4>; Implement a lexical, syntax and semantic analyzer; Bloom Taxonomy Level: <Apply>***

Here's a version of the CFG for a simplified subset of English

**Sentence -> NounPhrase VerbPhrase**

**NounPhrase -> Determiner Noun | NounPhrase PrepositionalPhrase**

**VerbPhrase -> Verb NounPhrase | VerbPhrase PrepositionalPhrase | Verb**

**PrepositionalPhrase -> Preposition NounPhrase**

**Determiner -> 'the' | 'a'**

**Noun -> 'cat' | 'dog' | 'man' | 'woman'**

**Verb -> 'chased' | 'caught'**

**Preposition -> 'in' | 'on' | 'with'**

**Sentence** represents a sentence composed of a noun phrase (subject) followed by a verb phrase (predicate).

**NounPhrase** A noun phrase is a phrase based around a noun, consisting of the noun and its modifiers. Examples: "the cat," "a dog," "a tall man," "the beautiful woman."

**VerbPhrase** A verb phrase is a phrase based around a verb, consisting of the verb and its complements and modifiers. Examples: "chased the cat," "caught a ball," "is running," "will sing a song."

Write a program that ask user to enter an English language sentence (for now enter series of tokens) and prints if the sentence is syntactically correct based on the provided CFG. Use recursive descent parsing algorithm to analyze the syntax and any errors in a basic You are not required to perform lexical analysis or tokenization, just input the tokens instead of words . [30 marks]

Example input:

<the,determiner ><dog,noun><caught, verb><a, determiner><cat, noun>

Output Required:

Complete stack trace of your algorithm and answer if the sentence is syntactically correct.

# Solutions:

```
#include<iostream>
using namespace std;
int offset = 0;

bool Sentence();
bool NounPhrase();
bool VerbPhrase();
bool PrepositionalPhrase();
bool Determinator();
bool Noun();
bool Verb();
bool Preposition();
bool NounPhraseDash();
bool VerbPhraseDash();
bool match(string);
//string input_tokens[] = {"the","dog","caught","the","cat"};
string input_tokens[] = {"the","cat","cased","a","dog"};

int main()
{
   if(Sentence())
      cout<<"Sentence is syntactically correct!"<<endl;
   else
      cout<<"Sentence is not syntactically correct!"<<endl;
}
bool Sentence()
{
   if(NounPhrase() && VerbPhrase()){
    cout<<"NounPhrase Verphrase"<<endl;
    return true;
   }
   return false;
}
bool NounPhrase()
{
   if(Determinator() && Noun() && NounPhraseDash())
   {
      cout<<"Determinator Noun NounPhraseDash"<<endl;
      return true;
   }
   return false;
}
bool NounPhraseDash()
{
   if(PrepositionalPhrase() && NounPhraseDash())
   {
      cout<<"PrepositionalPhrase NounPhraseDash"<<endl;
      return true;
   }
   else
      return true;
   return false;
}
```

```
bool VerbPhrase()
{
   if(Verb() && NounPhrase() && VerbPhraseDash())
   {
      cout<<"Verb NounPhrase VerPhraseDash"<<endl;
      return true;
   }
   else if(Verb() && VerbPhraseDash())
   {
      cout<<"Verb VerbPhraseDash"<<endl;
      return true;
   }
   return false;
}
bool VerbPhraseDash()
{
   if(PrepositionalPhrase() && VerbPhraseDash())
   {
      cout<<"PrepositionalPhrase VerbPhraseDash"<<endl;
      return true;
   }
   else
      return true;
   return false;
}
bool PrepositionalPhrase()
{
   if(Preposition() && NounPhrase())
   {
      cout<<"Preposition NounPhrase"<<endl;
      return true;
   }

   return false;
}
bool Determinator()
{
   if(match("the") || match("a"))
      return true;
   return false;
}
bool Noun()
{
   if(match("cat") || match("dog") || match("man") || match("women"))
      return true;
   return false;
}
bool Verb()
{
   if(match("chased") || match("caught"))
      return true;
   return false;
}
bool Preposition()
{
```

```
    if(match("in") || match("on") || match("with"))
        return true;
    return false;
}
bool match(string a)
{
    if(a == input_tokens[offset])
    {
        offset++;
        return true;
    }
    return false;
}
```