**University of Central Punjab**

**Faculty of Information Technology**



**Compiler Construction**

Project Phase # 2

Submission Before: 3:00PM - 30-05-2017

(Late will be penalty of deduction of 2 absolute marks per day)

UCP-Compiler:

This is a subset of C++ language. Description of the language as follow:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Detail | Note |
| 1 | Reserve words | constant, float, integer, break, continue, else, for, switch, void, case, char, do, if, return, union, while, |  |
| 2 | Identifiers | examples (val11u\_\_e, r2a\_te, etc) | ID must have “\_” and digit.  Cannot start with Digit. |
| 3 | Numbers (signed and un-signed) | example (223, 23.5, 0.45, .45, 45. , 45.0, -123, -23.5, -0.45, -.45, -45. , -45.0, etc) |  |
| 4 | Operators | < , > , != , <> , := , == , \* , + , / , - , >> , << , ++ , += , -= , -- , && , || , |& as XOR , ; , : |  |
| 6 | User input/output | In>>, out<< |  |
| 7 | Punctuations | [,{,(,),},],”,” |  |
| 8 | Comments | Handling single and multiline comments. |  |

**Grammar**

**Use Grammar provided in Course Material Folder. You have to find grammar which can accept following Code:**

intmainfunc(int I; int k)

{

int fn1 = 1,fn2 = 1 ;

inti=0;

int n;

floatfn=0;

if(n < 1)

{

fn1 = (fn1 + fn2)\*(fn1 – fn2);

}

else

{

while(i>= n)

{

fn = fn1 + fn2;

fn2 = fn1;

fn1 = fn;

}

}

}

intmainfunc(int I; int k)

{

floatfn=0;

if(n < 1)

{

fn1 = (fn1 + fn2)\*(fn1 – fn2);

}

while(i>= n)

{

fn = fn1 + fn2;

fn2 = fn1;

fn1 = fn;

}

}

# Assignment Description:

For this assignment,

1. You have to write a **Parser**for which accept above code.
2. **Parser** will get **Token** from scanner and built a parse tree.
3. Parser will built the parse tree using Predictive Parser (LL(1)) grammar.
4. **Panic Mode** approach will be implemented to output the syntax error.
5. This assignment includes following parts:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **PARTS** | **Output** | **Marks** |
| **1** | Convert grammar to LL(1) grammar. |  | 5 |
| **2** | Implement Parser using   1. LL1 Parsing Table or 2. Recursive Descent Technique using LL1 Grammar | Source Code Files | 35 |
|  | Total |  | 40 |
|  | **Absolute** |  | **10** |

# Rules:

1. This is an individual assignment. Each student has to submit his/her assignment work.
2. Group discussion is allowed but don’t share code and other part of assignment with other student.
3. Plagiarism is not tolerable in any of its form. Minimum penalty would be an ‘0’ marks in the project module.

# Evaluating Criteria:

1. Source code should reflect the detail given in documents (other parts).
2. A text file with valid source code will be input of the scanner and Token file will be output of the scanner tool.
3. A text file will show the productions in separate lines used in building the parse tree.
4. A text file show the errors generated from both scanner and parser.