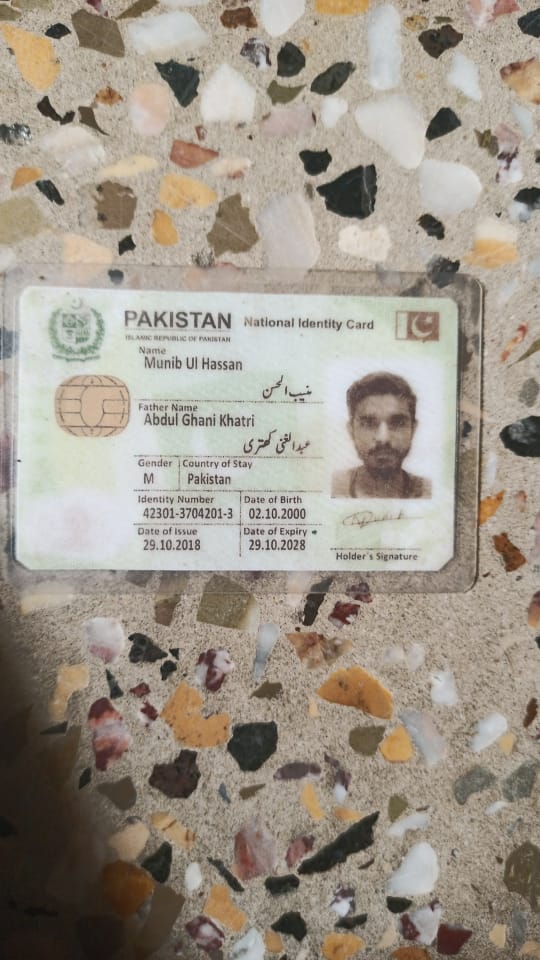
**Sir Syed University of Engineering & Technology**

ANSWER SCRIPT

|  |  |
| --- | --- |
| Date: | June 15,2021 |
| Roll Number: | CS19-037 |
| Section: | A |
| Name: | Munib ul Hassan |
| Course Name: | CS-310: Compiler Construction |
| Degree Program: | BSCS |
| Total number of pages being submitted: | 7 |

****

****

**Answer 01(a):**

**CODE:**

#include <stdio.h>

int main(void) {

int Per; printf("Enter Per : ");

scanf("%d",&Per);

if(Per >= 50)

printf("\nResult is pass");

else printf("\nResult is fail")

; return 0;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Marksheet1

{

class Program

{

static void Main(string[] args)

{

int r, m1, m2, m3, t;

float p;

string n;

Console.WriteLine("Enter Roll Number :");

r = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Student Name :");

n = Console.ReadLine();

Console.WriteLine("Mark of Subject1 : ");

m1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Mark of Subject2 : ");

m2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Mark of Subject3 : ");

m3 = Convert.ToInt32(Console.ReadLine());

t = m1 + m2 + m3;

p = t / 3.0f;

Console.WriteLine("Total : " + t);

Console.WriteLine("Percentage : " + p);

if (p >= 35 && p < 50)

{

Console.WriteLine("Grade is C");

}

if (p >= 50 && p <= 60)

{

Console.WriteLine("Grade is B");

}

if (p > 60 && p <= 80)

{

Console.WriteLine("Grade is A");

}

if (p > 80 && p <= 100)

{

Console.WriteLine("Grade is A+");

}

Console.ReadLine();

}

}

}

**CFG**

|  |
| --- |
| Program ⇾ Begin body End  Body ⇾ Stmnts  Stmnts ⇾ Stmnt\_List; stmnt | Stmnt\_List | Ɛ  Stmnt\_List ⇾ dec\_Stmnt | Print\_Stmnt | input\_Stmnt | assign\_Stmnt | loop\_Stmnt | if\_stmnt |cond\_Stmnt| func\_Smnt| return\_stmnt  dec\_Stmnt ⇾ dt var  func\_Smnt ⇾ dt var (dec\_stmnt) {stmnt ; return\_stmnt}| dt var ( ) {stmnt ; return\_stmnt}  return\_stmnt ⇾ return (var) | return (null) | Ɛ  Print\_Stmnt ⇾ Print val | Print String| Ɛ  Print val ⇾ Cout<< var ;  Print str ⇾ Cout << str  Str ⇾ “text”  Text ⇾ id | num | sp-char  input\_Stmnt ⇾ input val | Ɛ  assign\_Stmnt ⇾ var = E  E 🡪 EAE | id | num  A 🡪 + | - | \* | /  if\_stmnt ⇾ if (cond\_stmnt){stmnt}else {stmnt}  cond\_Stmnt ⇾ id relop id | id relop num  loop\_Stmnt ⇾ for\_loop  for\_loop ⇾ for (init ; cond\_stmnt ; inc\_dec){stmnts}  init ⇾ dt id = digit | id = digit  inc/dec ⇾ id++ | id-- | ++id | id + num | id – num  relop ⇾ < | > |<= | >= | != | ==  dt ⇾ int | void | float | char | double  id ⇾ var | var digit | var var  var ⇾ a|b|c  num ⇾ 0|1|2 … |9  sp-char ⇾ @ | / | \ | \* |

**Answer 01(b):**

E 🡪 EAE | id | num

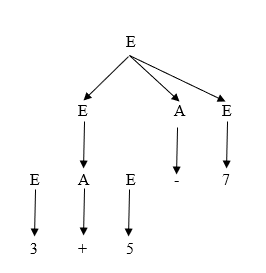
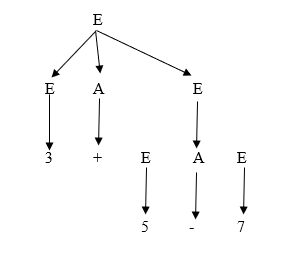
A **🡪** + | - | \* | /

This is anbigous clause

Example:

Input string 3 + 5 – 7

**BY PARSE TREE:**

Two parse tree can be generated by this input string, therefore it is ambiguous grammer.

**Answer 02**

|  |  |  |  |
| --- | --- | --- | --- |
| Nonterminal | Nullable | First | Follow |
| S | X | (, id |  |
| E | X | (, id | ), $ |
| E’ | X | +,37 | ), $ |
| T | X | (, id | +, 37 |
| T’ | X | \* , 37 | +, 37 |
| F | X | (, id | \* ,37 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | $ | + | 37 | \* | ( | ) | Id |
| S |  |  |  |  | S ::= E$ |  | S ::= E$ |
| E |  |  |  |  | E ::= TE’ |  | E ::= TE’ |
| E’ |  | E’ ::= + TE’ | E’ ::= 37 |  |  |  |  |
| T |  |  |  |  | T ::= FT’ |  | T ::= FT’ |
| T’ |  |  | T’ ::= 37 | T’ ::=\* FT’ |  |  |  |
| F |  |  |  |  | F ::= (E) |  | F ::= id |

E

T E’

F T’ + T E’

Id ℇ F T’ ℇ

Id ℇ

**Answer 03**

**PARSE TREE**

Z 🡪 AB

A 🡪 int

A 🡪 char

B 🡪 B1,id

B 🡪 id

Input String: Int 037

Z

A(4) B(5)

n6 addtype (n3,n5)

Int B1(7) 1 id(3)

n8 addtype(n2,n7)

B1(9) 1 id(2)

n10 add type (n1,n9)

id(1)

n1 = 0

n2 = 3

n3 = 7

n4 = int

n5 = n4

n6 = addtype(n3,n5)

n7 = n5

n8 = addtype(n2,n7)

n9 = n7

n10 = add type (n1,n9)

**Answer 04**

Rewriting the gramer

Start 🡪 Stmts n

Stmts 🡪 Stmts1 Detail

Stmts 🡪 Detail

Stmts 🡪 €

Detail 🡪 0

Detail 🡪 1

**SEMENTIC RULES:**

|  |  |
| --- | --- |
| **PRODUCTION** | **SEMENTIC RULE** |
| Start 🡪 Stmts n | Print {Stmts-val} |
| Stmts 🡪 Stmts1 Detail | Stmts.val := stmts1.val Detail.lexval |
| Stmts 🡪 Detail | Stmts.val := Detail.lexval |
| Stmts 🡪 € | Stmts.val := € |
| Detail 🡪 0 | Detail.lecxval := 0 |
| Detail 🡪 1 | Detail.lexval := 1 |

**Answer 05**

My RollNo is CS19-037

A = 0 + 3 + 7 = 10

My Date Of Birtth is October 02,2000

B = 0 + 2 + 1 + 0 + 2 + 0 + 0 + 0 = 5

1. pointer(Array(5….10 , Reals))
2. array [0-9][5-10]