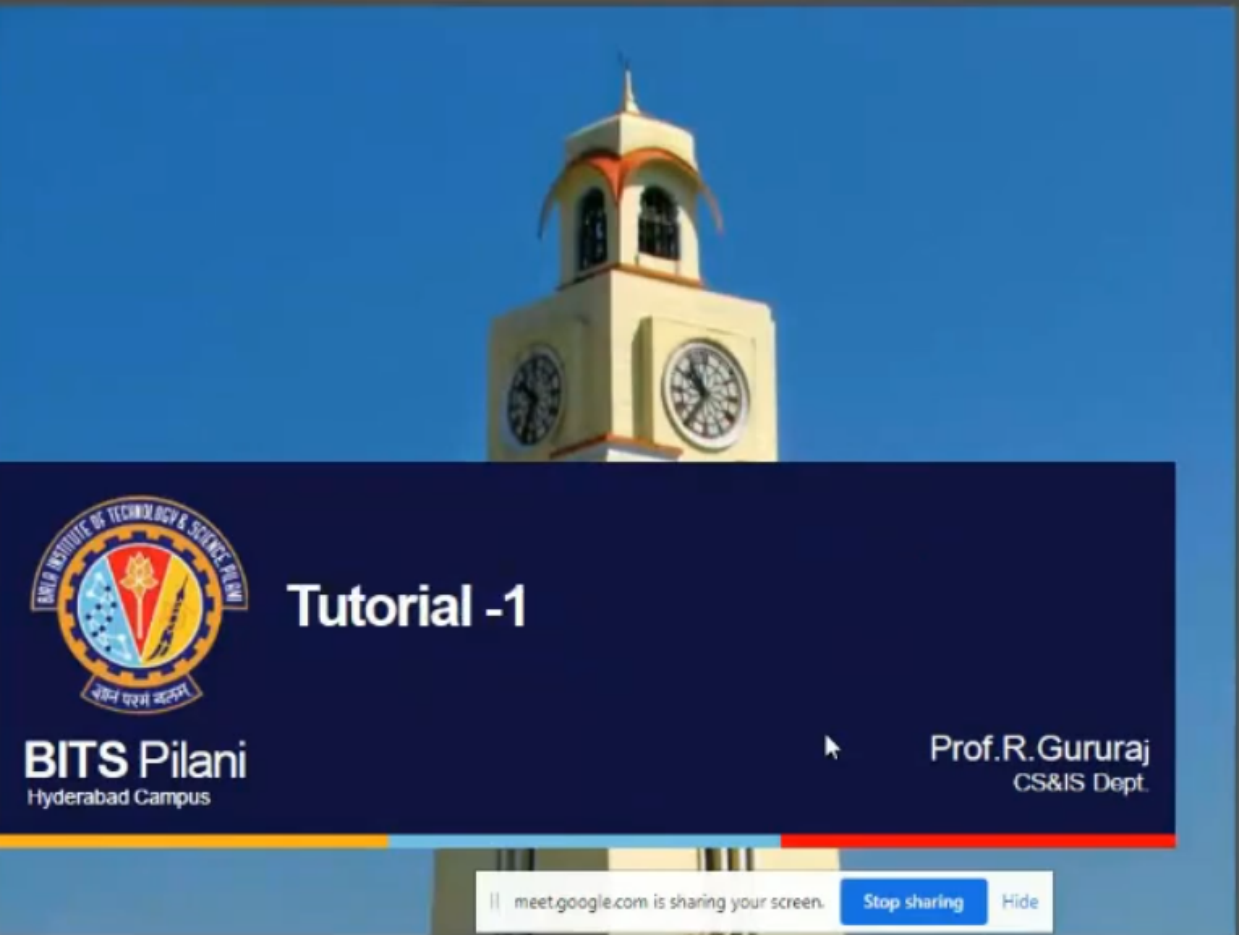
files.md

Tutorials

Tutorial 1



innovate achieve lead

Exercise-1

Solve the following problems using C and Java separately.

Write a program that would change every alternate '1' to '0' in a binary string given as input to a function.

Solution

```
#include<stdio.h>
#include<string.h>
void flip(char str[])
{
  int count=0, i=0;
  while(str[i]!='\0'){
    if(str[i]=='1'){
      count++;
    }
    if(count%2==0){
      str[i]='0';
    }
    i++;
  }
 printf("Output is %s\n", str);
}
```

```
int main()
{
  char bits[20];
  printf("Enter a binary pattern: \n");
  scanf("%s", bits);
  printf("Entered binary is %s\n", bits);
  flip(bits);
  return 0;
}
```


Solution

```
import java.util.Scanner;
class Bits{
public static void flip(char[] str) {
    int count = 0,i=0;

  while(i<str.length){
    if(str[i]=='1'){
      count++;
    }
    if(count%2==0){
      str[i]='0';
    }
    i++;
  }
 System.out.println("Output is :" + new String(str));
}
```

```
 public static void main(String args[]){
    System.out.println("Enter a binary
pattern: ");
    Scanner s = new Scanner(System.in);
    char[] bits = s.next().toCharArray();

    flip(bits);
   }
}//end of class
```

Exercise-2

Solve the following problems using C and Java separately.

Write a program where we need to store data pertaining to students in an array. Each student will have id int, name character string, cgpa float as attributes.

We should have a function/method addStudent() that takes the above three values as args and create a student entity (design appropriately) and add to an array that stores the student entities. Method/function printStudents()

must print the details of each student available in the array. Array may be defined to hold max 5 students.

Hint: For C implementation use Structures, for Java use class concept.

One possible Solution

```
#include <stdio.h>
#include<string.h>
int count = 0;
struct student {
   int id;
   char Name[50];
   float marks;
} s[5];
```

```
void printStudent(){
   int i;
   for (i = 0; i < count; ++i) {
      printf("\nRoll number: %d\n", s[i].id);
      printf("First name: ");
      puts(s[i].Name);
      printf("Marks: %.1f", s[i].marks);
      printf("\n");
   }
}
```

```
void addSudent(int id, char name[],float marks){
   s[count].id = id;
   strcpy(s[count].Name,name);
   s[count].marks = marks;
   count++;
}
```

One possible Solution cont...

```
int main() {
    int i,a;
    char b[50];
    float c;
    printf("Enter information of students:\n");

    for (i = 0; i < 5; ++i) {
        printf("\n Enter id number:\n");
        scanf("%d", &a);
        printf("Enter first name: ");
        scanf("%s", b);
        printf("Enter marks: ");
        scanf("%f", &c);
        addSudent(a,b,c);
    }
    printf("Displaying Information:\n\n");
    printStudent();

    return 0;
}
```


Java Solution

```
class Student{
        int id; String name; float cgpa;
    Student(int i, String n, float cg)
        {id=i; name=n; cgpa=cg;   }
}
class StudentDemo
{  public static void main(String args[])
   {    Student stds[]= new Student[5];
            stds[0]=addStudent(101,"Raj", 8.90f);
            stds[1]=addStudent(103,"Kumar", 7.50f);
            stds[2]=addStudent(106,"Tom", 6.50f);
            stds[3]=addStudent(108,"Sushma", 8.35f);
            stds[4]=addStudent(110,"Phani", 6.10f);
            printStudent(stds);
    }
    static Student addStudent(int sid, String sn, float scg)
    {   Student st=new Student(sid, sn, scg); return st;}

     static void printStudent(Student s[])
     {
       for (int c=0;c<s.length;c++)
       { System.out.println("ID: "+s[c].id);
         System.out.println("Name : "+s[c].name);
         System.out.println("Grade: "+s[c].cgpa);
       }
     }
}
```

C

```c
#include<stdio.h>
int main()
{
    int a, b, c;
    printf("Enter two
       numbers to add\n");
    scanf("%d%d", &a, &b);
    c = a + b;
    printf("Sum of the
      numbers = %d\n", c);
    return 0;
}
```

JAVA

```java
import java.util.Scanner;
class AddNumbers
{
    public static void main(String args[])
    {
        int x, y, z;
        System.out.println("Enter two Nos");
        Scanner in = new Scanner(System.in);
        x = in.nextInt();
        y = in.nextInt();
        z = x + y;
        System.out.println("Sum = " + z);
    }
}
```


Sum of two numbers

Haskell

```
main:: IO()
main = do
        putStrLn "Insert the 1st no:"
        one <- getLine
        putStrLn "Insert the  2nd no:"
        two <- getLine
        putStrLn "The result is:"
        print (read one + read two)
```

PHP

```php
<?php
        $x=15;
        $y=30;
        $z=$x+$y;
        echo "Sum: ",$z;

?>
```

PYTHON

```
a = int(input("enter 1st no: "))
b = int(input("enter 2nd  no: "))
sum = a + b
print("sum:", sum)
```

RUBY

```
print "enter number 1 : ";
val1 = gets;
print "enter number 2 : ";
val2 = gets;
print "Answer : ",(val1.to_i + val2.to_i), "\n";
```


Why all Languages are not popular?

- It takes time to learn a programming language, and the programming community will not entertain a new language with no real uses.
- Too much-sophisticated features that confuse developers.
- Poor Documentation & Community-support
- When a programmer gets experienced, they are more committed to solving problems rather than learning new languages.

Revising a programming Language?

Revision process will continually add new features, so that the language grows more and more complex.

Compounding the problem is the unwillingness to remove obsolete features, because of existing software.

Backward Compatibility.

User-Adaptability.

Tutorial 2

Tutorial -2

BITS Pilani
Hyderabad Campus

Prof.R.Gururaj
CS&IS Dept

Two kinds of commenting approaches used in C and Java:

innovate achieve lead

1. Multiple-line comments- Give Pros & Cons
2. Single-line comments- Give Pros & Cons

1. Delimiters are used on both ends (multiple-line comments)

The main disadvantage of using paired delimiters for comments is that it results in diminished reliability. It is easy to accidentally leave off the final delimiter, which extends the comment to the end of the next comment, effectively removing code from the program.

The advantage of paired delimiters is that one can comment out areas of a program.

2. Delimiter marks only the beginning of the comment (single-line comments).

The disadvantage of using only beginning delimiters is that they must be repeated on every line of a block of comments. This can be tedious and therefore error-prone.

The advantage is that you cannot make the mistake of forgetting the closing delimiter.

Distinguish between uppercase and lowercase letters in user-defined names.

(1) Variable identifiers may look different than identifiers that are names for constants, such as the convention of using uppercase for constant names and using lowercase for variable names in C.

(2) Catenated words as names can have their first letter distinguished, as in TotalWords. (it is better to include a connector, such as underscore)

An Advice:

int SUM, sum; /*This may become confusing in later lines of code*/

int sum1, sum2; /*this is more readable and easy to associate and remember throughout the program */

We know that Java uses a pair of curly brace to mark the end of all compound statements.

Give Advantages & Disadvantages of this approach.

innovate achieve lead

"Java uses a right brace to mark the end of all compound statements"

The argument for using the right brace to close all compounds is simplicity—a right brace always terminates a compound.

The argument against it is that when you see a right brace in a program, the location of its matching left brace is not always obvious, in part because all multiple-statement control constructs end with a right brace.

Hence it is felt that use of IF .... END IF kind of syntax makes it more readable.

Example:

Language Implementation Methods (Sec.7 of Ch.1 in T1)

1. Compilation
2. Interpretation
3. Hybrid approach
4. Pre-Processors

Compilers and Interpreters are known as language processors.

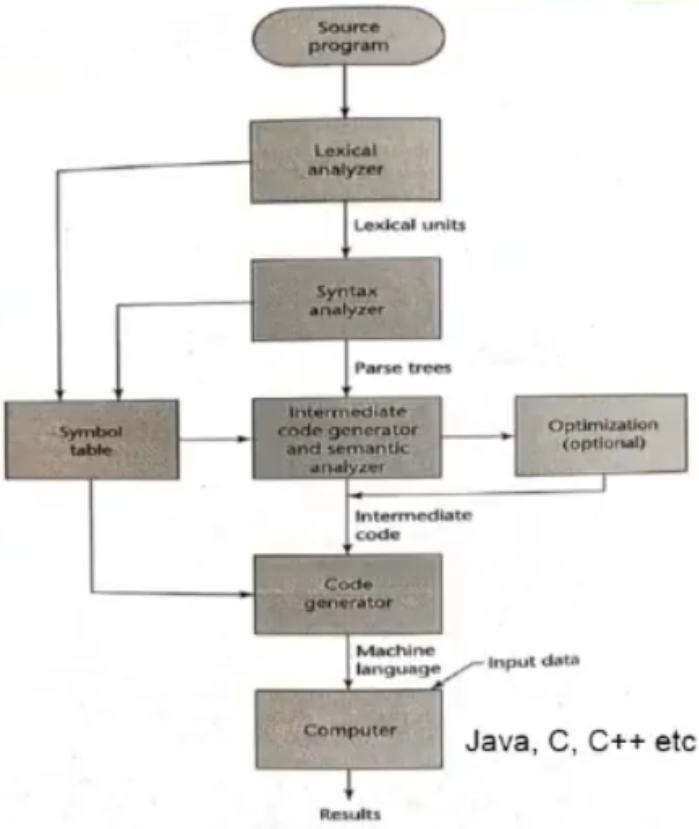
Compiler as a Language Processors

Compiler: It is a program that reads a program written in one language (source language) and translates it into an equivalent program in another language(target language).
Ex: C, C++, Ada, COBOL

It also reports errors in the source program.

It is a Software system for translation.

Compilation



token
<token-name, value>

1	position	...
2	initial	...
3	rate	...

SYMBOL TABLE

<id,1>
↓
Symbol table entry

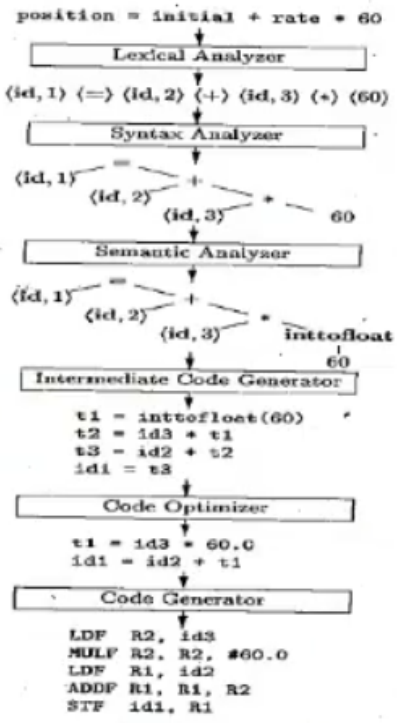
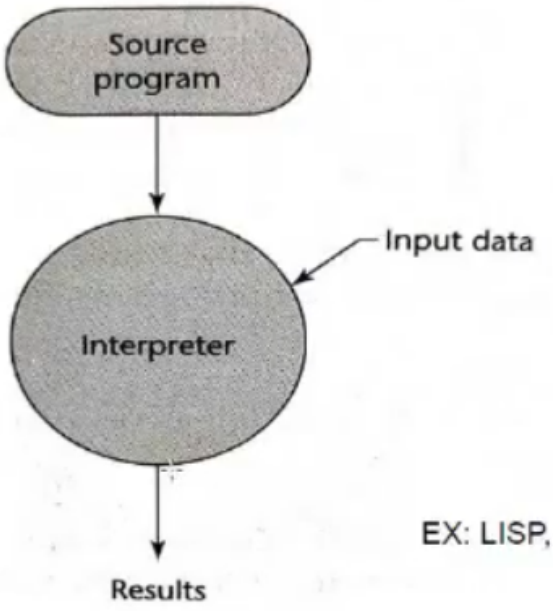


Figure 1.7: Translation of an assignment statement

Interpreter as a Language Processor

Interpreter: Instead of producing a target program, it directly executes the operations specified in the source program on inputs supplied by the user and produces output.
With this approach, the programs are interpreted by another program called an interpreter. No translation required.
The interpreter program acts as a SW simulator of machine whose fetch-decode-execute deals with high-level-program statements rather than machine instructions.
This SW simulation provides a VM for the language.
Debugging is easy.
But slow because interpretation must happen every time you execute the program. Ex: LISP, SNOBOL, Scripting languages like PHP, Java Script etc.
A Hybrid approach: In java the byte code is interpreted by JVM.

Interpretation



EX: LISP, JavaScript, PHP

Hybrid Approach

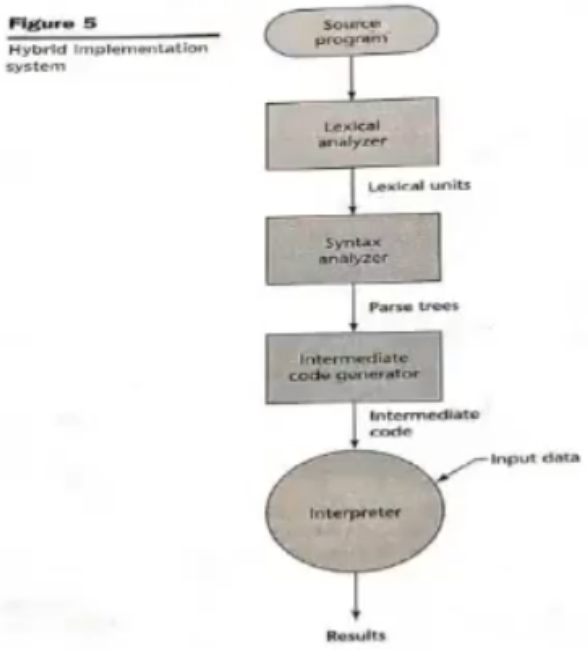


Figure 5
Hybrid implementation system

Ex: Java

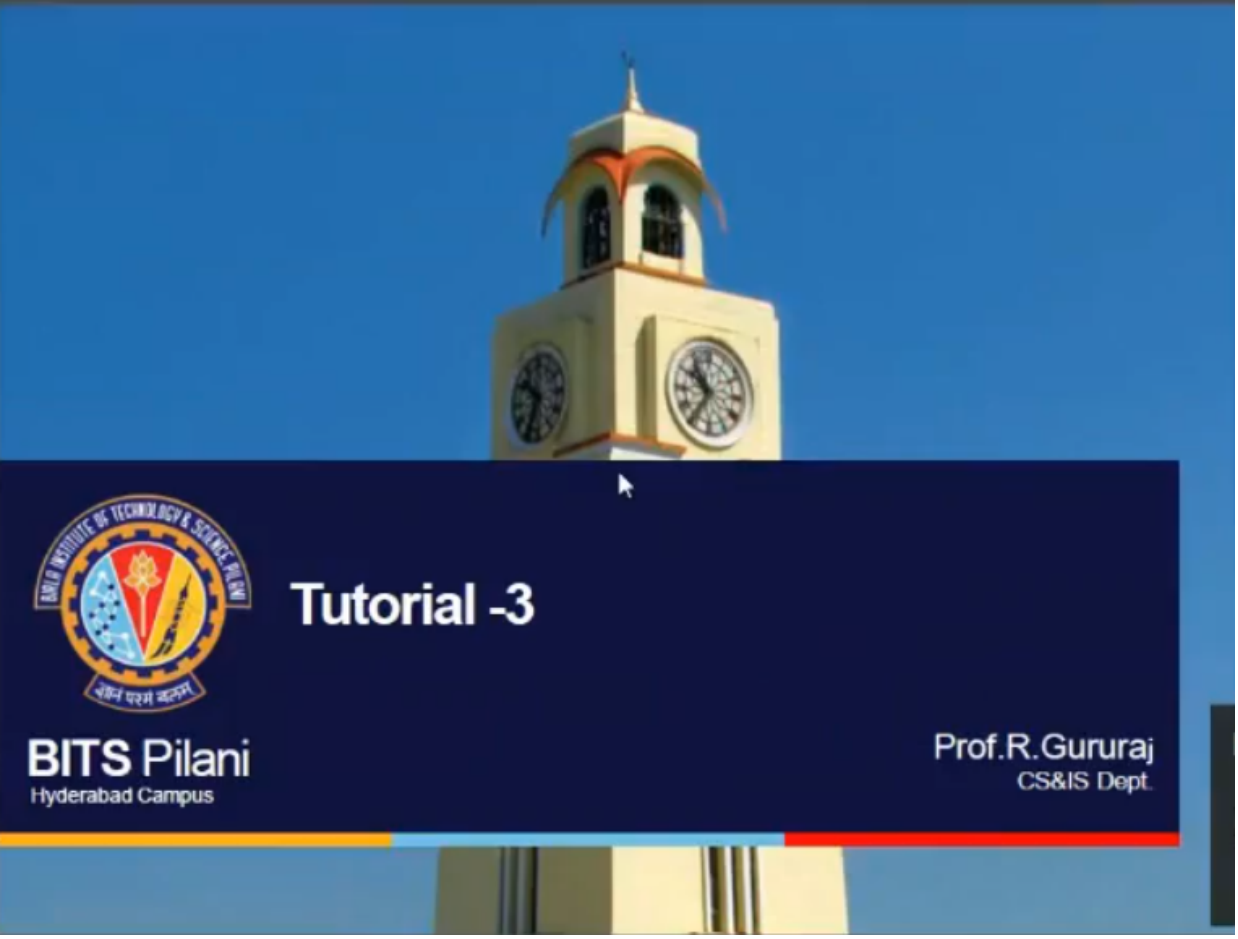
JIT Compilation:

A Just-in-Time (JIT) implementation system initially translates programs to an intermediate language. Then, during execution, it compiles intermediate language methods into machine code when they are called. The machine code version is kept for subsequent calls. JIT systems are now widely used for Java programs. Also, the .NET languages are all implemented with a JIT system.

Preprocessor: Is a program that processes the source code immediately before the code is compiled. Preprocess

A preprocessor is a program that processes a program immediately before the program is compiled. Preprocessor instructions are embedded in programs. The preprocessor is essentially a macro expander. Preprocessor instructions are commonly used to specify that the code from another file is to be included. For example, the C preprocessor instruction

Tutorial 3



Tutorial -3

BITS Pilani
Hyderabad Campus

Prof.R.Gururaj
CS&IS Dept.

innovate achieve lead

Example for Orthogonality

```
A   Reg1, memory_cell
AR  Reg1, Reg2
```

Assembly language for IBM mainframe

where Reg1 and Reg2 represent registers. The semantics of these are

```
Reg1 ← contents(Reg1) + contents(memory_cell)
Reg1 ← contents(Reg1) + contents(Reg2)
```

The VAX addition instruction for 32-bit integer values is

VAX series minicomputer

```
ADDL  operand_1, operand_2
```

whose semantics is

```
operand_2 ← contents(operand_1) + contents(operand_2)
```

Hence VAX language is considered to be more orthogonal, as it has less restrictions.

Exercise-1

Give a few examples of lack of orthogonality in the design of C?

Answer

1. Structures (but not arrays) may be returned from a function.
2. An array can be returned if it is inside a structure.
3. A member of a structure can be any data type (except void), or the structure of the same type.
4. An array element can be any data type (except void).
5. Everything is passed by value (except arrays).
6. Void can be used as a type in a structure, but a variable of this type cannot be declared in a function.

Exercise-2

What does it mean for a program to be reliable?

Answer

A program is said to be reliable if it performs to its specifications under all conditions.

Ex:

-Type checking.

-Exception handling

-Aliasing may cause issues

Exercise-3

What is aliasing? What are the bad effects of it?

Answer

Aliasing is having two or more distinct names that can be used to access the same memory cell. Others languages greatly restrict aliasing to increase their reliability.

Example may be given.

Exercise-4

innovate achieve lead

What is exception handling?

Answer

innovate achieve lead

Exception handling is ability of a program to intercept run-time errors (as well as other unusual conditions detectable by the program), take corrective measures, and then continue is an obvious aid to readability.

Prof.R.Gururaj CSF301 PPL Tutorial BITS Pilani, Hyderabad Campus

Exercise-5

innovate achieve lead

Why is readability important to writability?

Answer

innovate achieve lead

Because most of the language characteristic that affect readability also affects writability. This follows directly from the fact that the process of writing a program requires the programmer frequently to reread the part of the program that is already written.

Exercise-6

What have been the strongest influences on programming language design over the past 50 years?

Answer

- Structured-programming movement.
- Top-down design and stepwise refinement.
- A shift from procedure-oriented to data-oriented program design methodologies began.
- Supporting data abstraction.
- Object-oriented design.

Answer

The primary programming language deficiencies that were discovered were:

1. incompleteness of type checking and
2. inadequacy of control statements.

Exercise-8

What is an example of two language design criteria that are in direct conflict with each other?

Answer

1. Reliability and cost of execution.

   Ex: Type checking

2. Writability and Reliability .

   Ex: Pointers.

3. Readability and expressivity

   Ex: a huge amount of computation specified by a small program. Easy to write by difficult to read.

Exercise-9

What are the three general methods of implementing a programming language?

Answer

Three general methods of implementing a programming language are:

1. Compilation
2. pure interpretation, and
3. hybrid implementation.

Exercise-10

What role does the symbol table play in a compiler?

Answer

Serves as a database for the compilation process. A symbol table is a compile time data structure that is used by the compiler to collect and use information about the source program constructs, such as variables, constants, functions, etc.

The symbol table helps the compiler in determining and verifying the semantics of given source program.

The information in the symbol table is entered in the lexical analysis and syntax analysis phase, however, is used in later phases of compiler (semantic analysis, intermediate code generation, code optimization, and code generation).

Exercise-11

What are the advantages in implementing a language with a pure interpreter?

Answer

One of the advantages of implementing a language with a pure interpreter is- easy implementation of many source-level debugging operations, because all run-time errors can refer to the source-level units. An example of this is an array index is found to be out of range.

Exercise-12

What arguments can you make against the idea of a single language for all programming domains?

Answer

Some arguments against having a single language for all the programming domains are:

The language would necessarily be huge and complex.

The compilers would be expensive and costly to maintain.

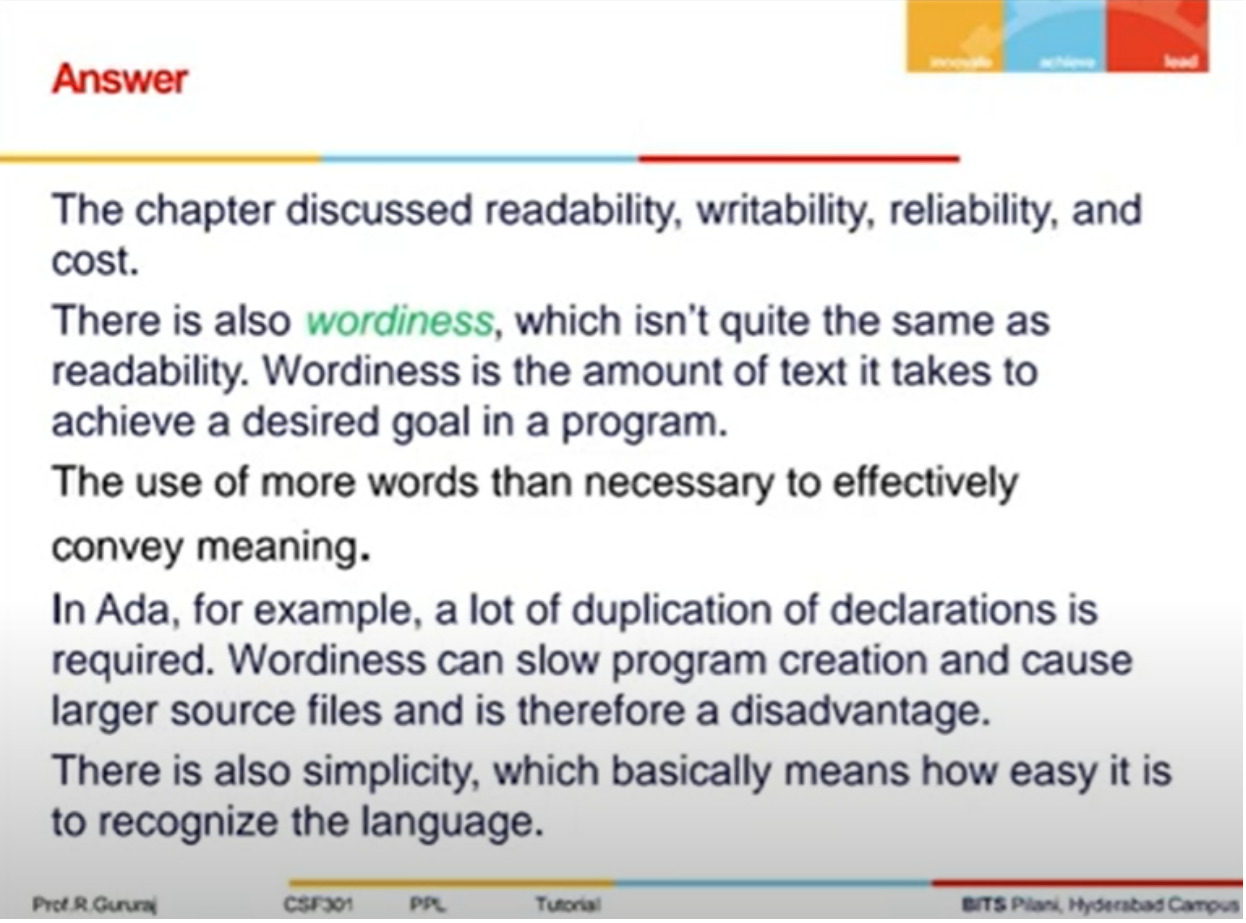
The language would probably not be very good for any programming domain, either in compiler efficiency or in the efficiency of the application area,

The language would include many unnecessary and confusing features and constructs.

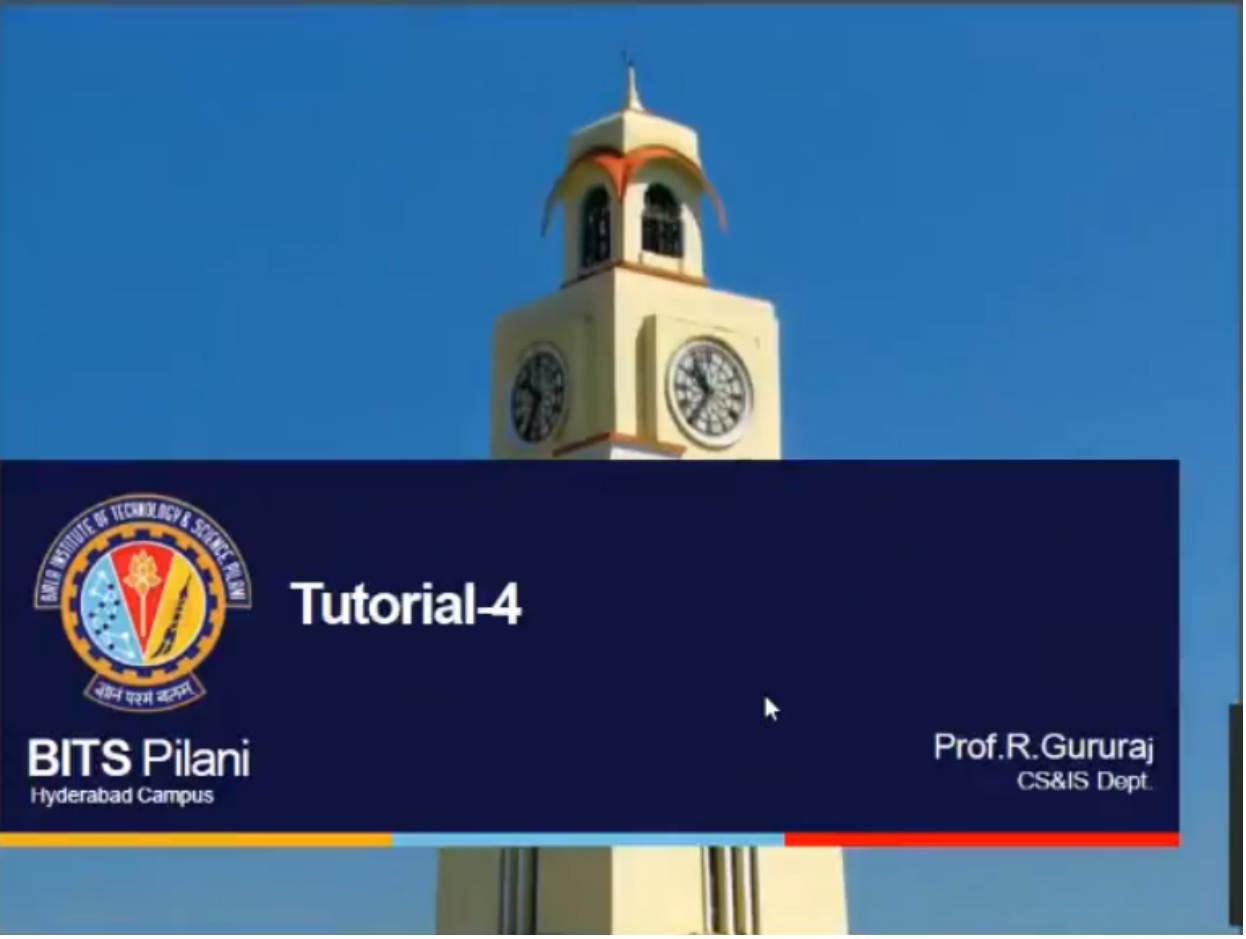
Different users would learn different subsets, making maintenance difficult.

Exercise-13

Name and explain another criterion by which languages can be judged.


Answer
The chapter discussed readability, writability, reliability, and cost.
There is also wordiness, which isn't quite the same as readability. Wordiness is the amount of text it takes to achieve a desired goal in a program.
The use of more words than necessary to effectively convey meaning.
In Ada, for example, a lot of duplication of declarations is required. Wordiness can slow program creation and cause larger source files and is therefore a disadvantage.
There is also simplicity, which basically means how easy it is to recognize the language.
Prof.R.Gururaj CSF301 PPL Tutorial BITS Pilani, Hyderabad Campus

Tutorial 4



innovate achieve lead

Q1.What are the three original goals of the ALGOL design committee, and in your opinion, which was most difficult to achieve at that time?

Q1.Answer: Three goals of the ALGOL design committee .

A. The syntax of the language should be as close as possible to standard mathematical notation, and programs should be readable with little further explanation.

B. It should be possible to use the language for description of algorithms in printed publications.

C. Programs in the new language must be mechanically translatable into machine code.

Among the above three, the First goal was the most difficult to achieve at that time, because it competed with the another existing language (Fortran) for scientific applications to be the universal language of its application area at that time.

Q2. Describe in detail the three most important reasons, in your opinion, why ALGOL 60 did not become a very widely used language.

Q2. **Answer:**

A. Some of the features of ALGOL 60 turned out to be too flexible, they made understanding difficult and implementation inefficient.

B. The lack of input and output statements in the language was another major reason for its lack of acceptance. Implementation-dependent input/output made programs difficult to port to other computers.

C. Lack of support by IBM.

Q3. Outline the major motivation of IBM in developing PL/I.

Q3.Answer:

The main motivation for the development of PL/I was to provide a single tool for computer centres that must support both scientific and commercial applications.

IBM believed that the needs of the two classes of applications were merging, at least to some degree.

They felt that the simplest solution for a provider of systems, both hardware and software, was to furnish a single hardware system running a single programming language that served both scientific and commercial applications.

Q4. What are the arguments both for and against the idea of a typeless language?

Q4. **Answer:**

The argument for typeless languages is their great flexibility for the programmer. Literally any storage location can be used to store any type value. This is useful for very low-level languages used for systems programming. The drawback is that type checking is impossible, so that it is entirely the programmer's responsibility to insure that expressions and assignments are correct.

Q5. Are there any logic programming languages other than Prolog?

Q5.Answer: Yes, we have.

- Mercury,
- ALF(Algebraic Logic Functional programming language),
- Fril etc.

Q6. What is your opinion of the argument that languages that are too complex are too dangerous to use, and we should therefore keep all languages small and simple?

Q6. **Answer:**

Simple programming languages are easier to learn and also with their small numbers of feature programmer can solve almost all problems by combining them.

If it is too complex programmers may not learn and use all of them.

Q7. Explain two reasons why pure interpretation is an acceptable implementation method for several recent scripting languages.

Q7.Answer:

1.One situation in which pure interpretation is acceptable for scripting languages is when the amount of computation is small, for which the processing time will be negligible.

2. No compilation means the time from editing code to testing the app can be diminished

3. No need to generate binaries for multiple architectures because the interpreter will manage the architecture abstraction, this supports embedding code into HTML and execute at remote machine (in browser) whose architecture is not known.

Hence best suited for Web programming.

Q8. Why, in your opinion, do new scripting languages appear more frequently than new compiled languages?

Q8. **Answer:**

New scripting languages may appear more frequently than new compiled languages because they are often smaller and simpler and focused on more narrow applications, which means their libraries need not be nearly as large.

innovate achieve lead

Q9. Give a brief general description of a markup / programming hybrid language.

Prof.R.Gururaj CSF301 PPL Tutorial **BITS** Pilani, Hyderabad Campus

Q9.Answer: A markup/ programming hybrid language is a markup language which contains programming actions also. That is, it can include control flow and computations also.

(1) A markup language contains processing instructions (called tags) in documents so that they are accurately presented.

(2) The language specifies code for formatting, definition and presentation of text within a text file.

(3) They also include additional descriptive information called metadata about the complex data. Examples for mark up languages are XML, HTML and SGML (Standard Generalized Markup Language) of which the most used one is HTML for presenting web pages in web browsers.

Q10. Brief on Garbage collection in Java.

Garbage Collection in Java

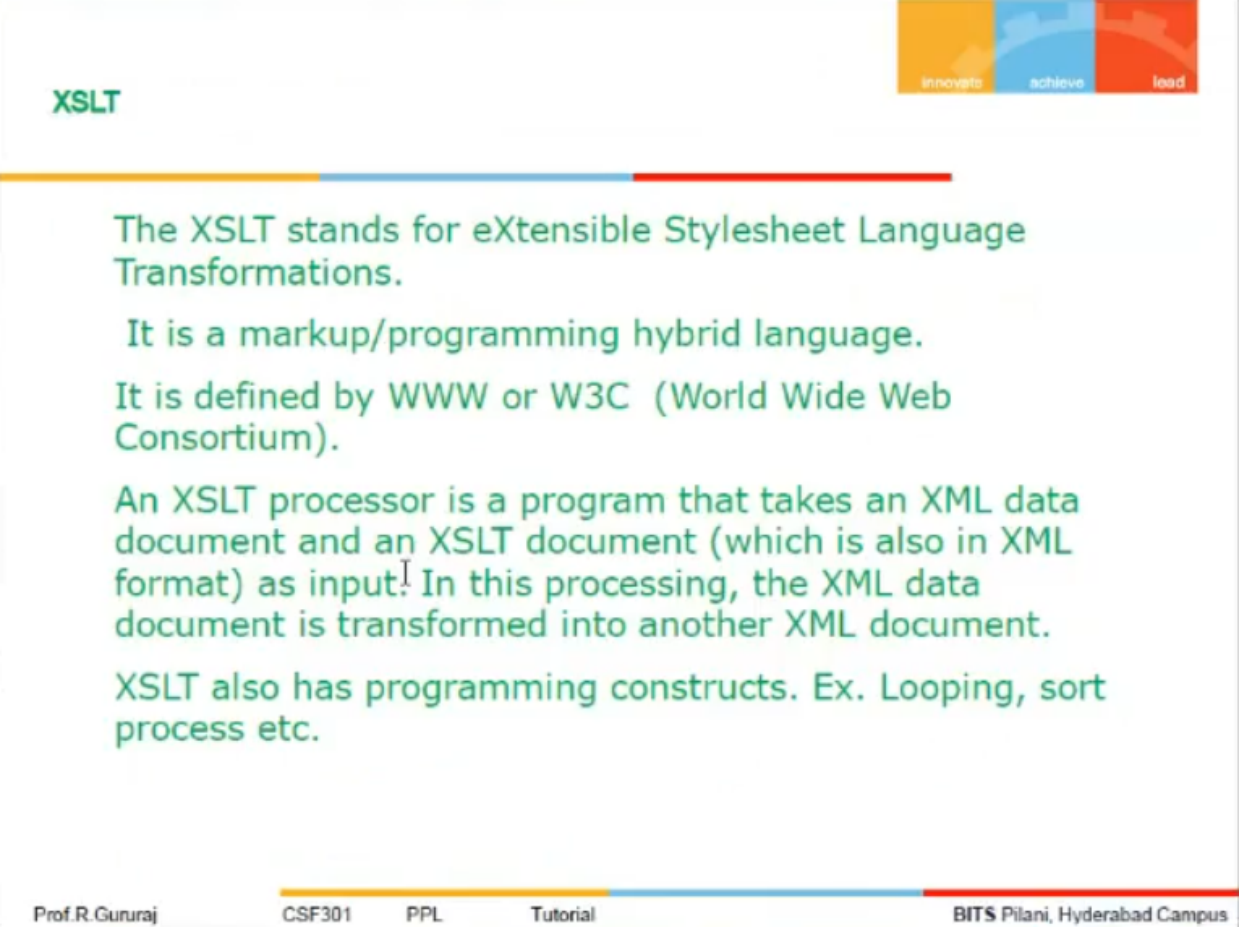
- In Java we don't destroy objects that are created.
- Java has a mechanism called '***garbage collection***'.
- It is daemon thread.
- It reclaims space occupied by unused objects.

Java applications obtain objects in memory as needed. It is the task of garbage collection (GC) in the Java virtual machine (JVM) **to automatically determine what memory is no longer being used by a Java application and to recycle this memory for other uses**.

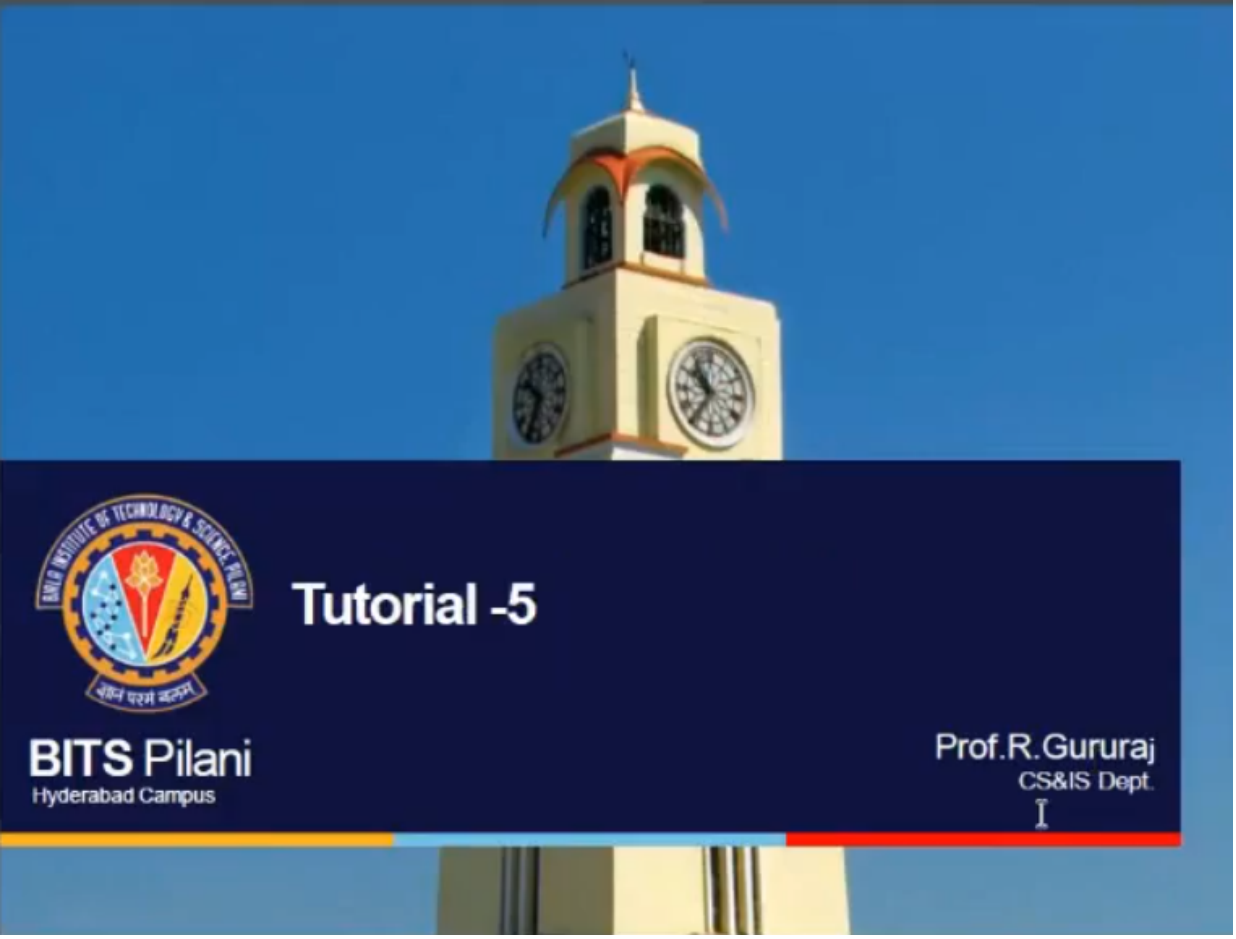
You can call Garbage Collector explicitly, but JVM decides whether to process the call or not. Ideally, you should never write code dependent on call to garbage collector.

Garbage collector is a daemon thread (low priority thread running in the background).

Q11. Brief on XSLT.


innovate
achieve
lead
XSLT
The XSLT stands for eXtensible Stylesheet Language Transformations.
It is a markup/programming hybrid language.
It is defined by WWW or W3C (World Wide Web Consortium).
An XSLT processor is a program that takes an XML data document and an XSLT document (which is also in XML format) as input. In this processing, the XML data document is transformed into another XML document.
XSLT also has programming constructs. Ex. Looping, sort process etc.
Prof.R.Gururaj
CSF301
PPL
Tutorial
BITS Pilani, Hyderabad Campus

Tutorial 5



innovate achieve lead

Question-1

Construct a CFG for all integers with sign.

Ex: -17; +23; etc.

And show how -17 is derived.

CFG <S>→ <sign> <integer>

<sign> → – | +

< integer> → <digit> <integer> | <digit>

<digit> → 0 | 1 |2 |.....|9

Ex: to derive " – 17 "

S→ <sign> <integer> → – <integer> → – <digit><integer> → – 1 <integer> → – 1 <digit> → – 17

Question-2

Consider the following CFG.

S→0Y | 01

Y→XY1 | 0

X→ 0XY | 0

Set of Non-terminals={S, X, Y}

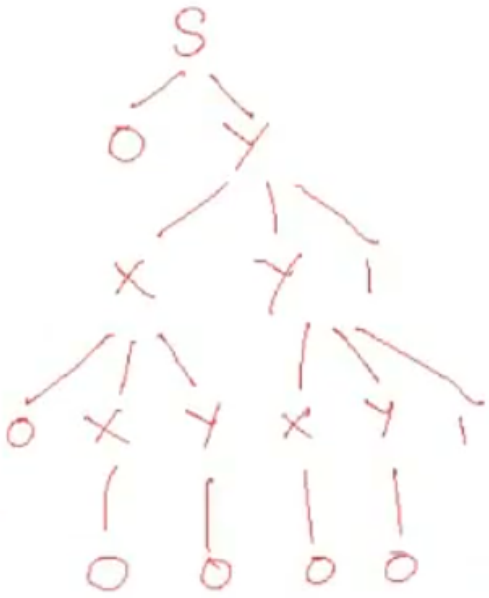
Set of terminals={0,1}

Starting Non-terminal=S.

Give two distinct derivations for 00000011.

Question-2 Solution

Consider the string 00000011



QUESTION 3

Look at the following Grammar.

S→ aScB

S→ A | b

A→ cA | c

B→ d | A

Set of Non-terminals={S, A, B} Set of terminals={a, b, c, d}

Starting Non-terminal=S.

Which of the following sentences can be generated by the above grammar?

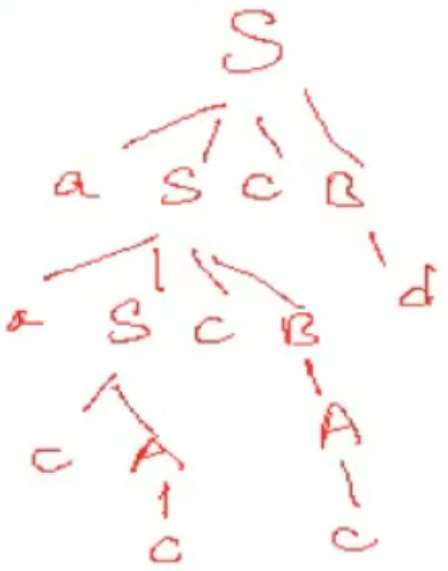
(i) aacccccd

(ii) acccbccd

(iii) accccd

Only (i) and (iii) can be generated.

(i) aaccccd



(iii) accccd

QUESTION 4

Give RE for Language Consisting of Strings on $\Sigma = \{a, b\}$ where every String Contains at least 2 as

Regular Expression :

$$(a \cup b)^* a (a \cup b)^* a (a \cup b)^*$$

Note: other valid Regular Expressions may be possible.

QUESTION 5

Give RE for String with even number of zeros followed by odd number of 1s.

$(00)^* (11)^* 1$

Note: other valid Regular Expressions may be possible.

QUESTION 5

(i) $S \rightarrow aS \mid aSbS \mid c$ (here, $\sum = \{a,b,c\}$)

1. Give two distinct Derivation for '*aacbc*', such that the parse trees are not identical.

Analyze why the parse trees are not identical, though the derivations are distinct.

$S \rightarrow aS$

$S \rightarrow aSbS$

$S \rightarrow c$

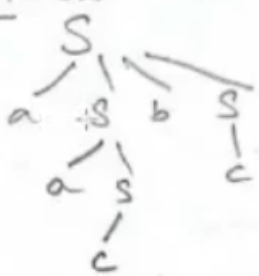
String 'aacbc'

Derivation 1: D1: $S \rightarrow a\underline{S}bS \rightarrow aa\underline{S}bS \rightarrow aacb\underline{S} \rightarrow aacbc$

Derivation 2: D2: $S \rightarrow aSb\underline{S} \rightarrow aSbc \rightarrow aaSbc \rightarrow aacbc$

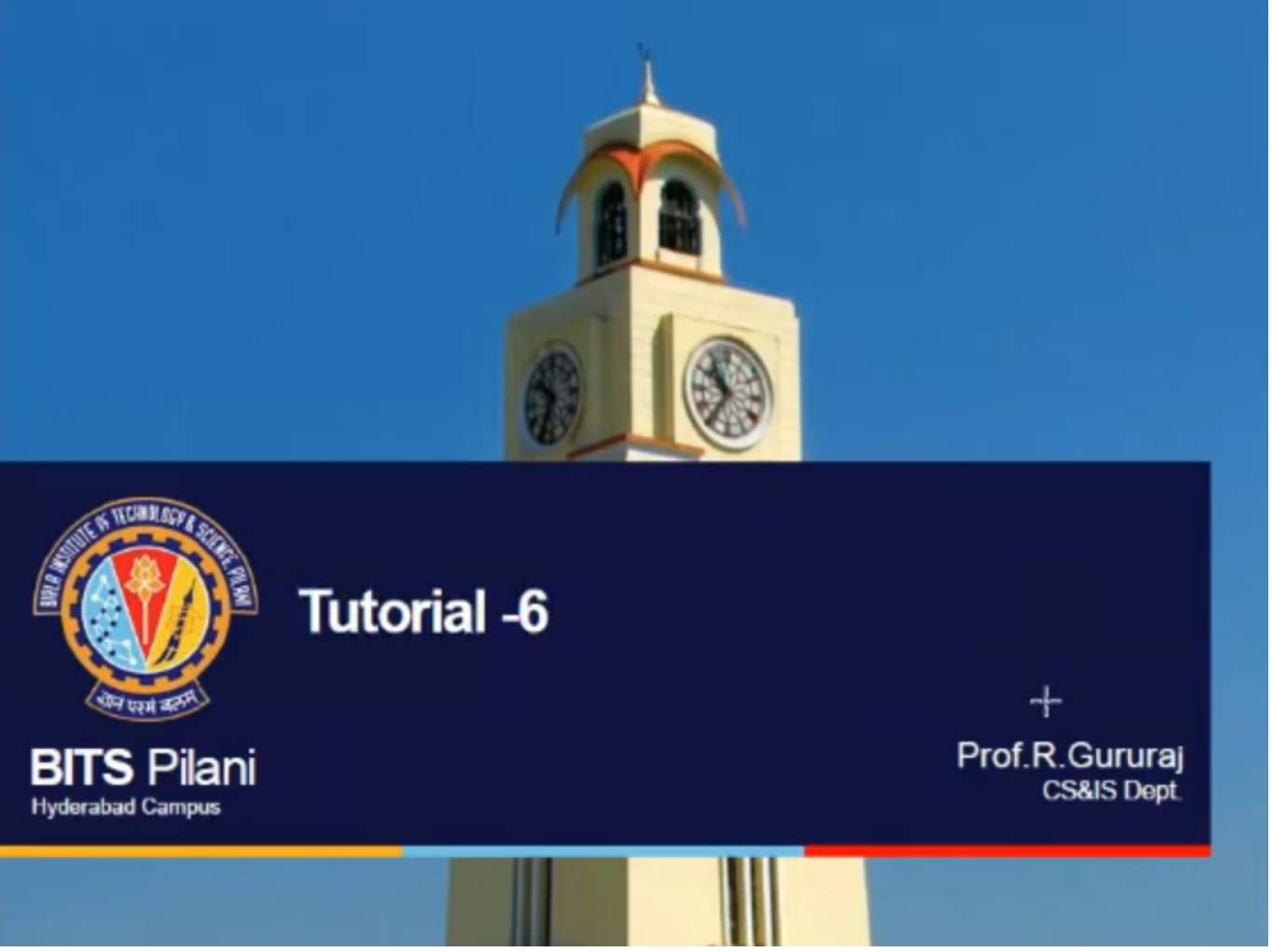
parse tree for D1:

parse tree for D2:



Why ?: Both are identical. because replacement of a NT is same but the order alone is different.

Tutorial 6



Tutorial -6

BITS Pilani
Hyderabad Campus

Prof.R.Gururaj
CS&IS Dept.

innovate achieve lead

Exercise – 1

Prove that the following Grammar is ambiguous.

S→ A

A→ A+A | M

M→ a | b |c

Exercise-1 : Solution.

Consider the string "a+b+c"

S→A→A+A→A+A+A→M+A+A→a+A+A→a+M+A→a+b+M→a+b+c

S→A→A+A→M+A→a+A→a+A+A→a+M+A→a+b+M→a+b+c

We have two LMDs hence it is ambiguous.

Exercise – 2

Give CFG for the language containing strings of the form={ $a^n b^{2n} a^m \mid n \geq 1$ and $m \geq 0$}.

Exercise-2 : Solution.

$$G = \{ S \rightarrow aAbbB$$
$$A \rightarrow aAbb \mid e$$
$$B \rightarrow aB \mid e \}$$

Convert the following BNF to EBNF

<program> → **begin** <stmt_list> **end**

<stmt_list> → <stmt>
| <stmt> ; <stmt_list>

<stmt> → <var> = <expression>

<var> → A | B | C

<expression> → <var> + <var>
| <var> - <var>
| <var>

Exercise-3 : Solution.

<program> → begin <stmt_list> end
<stmt_list> → <stmt> { ; <stmt>}
<stmt>→ <var>=<exression>
<var>→ A | B | C
<expression>→ <var> (+|-) <var> | <var>

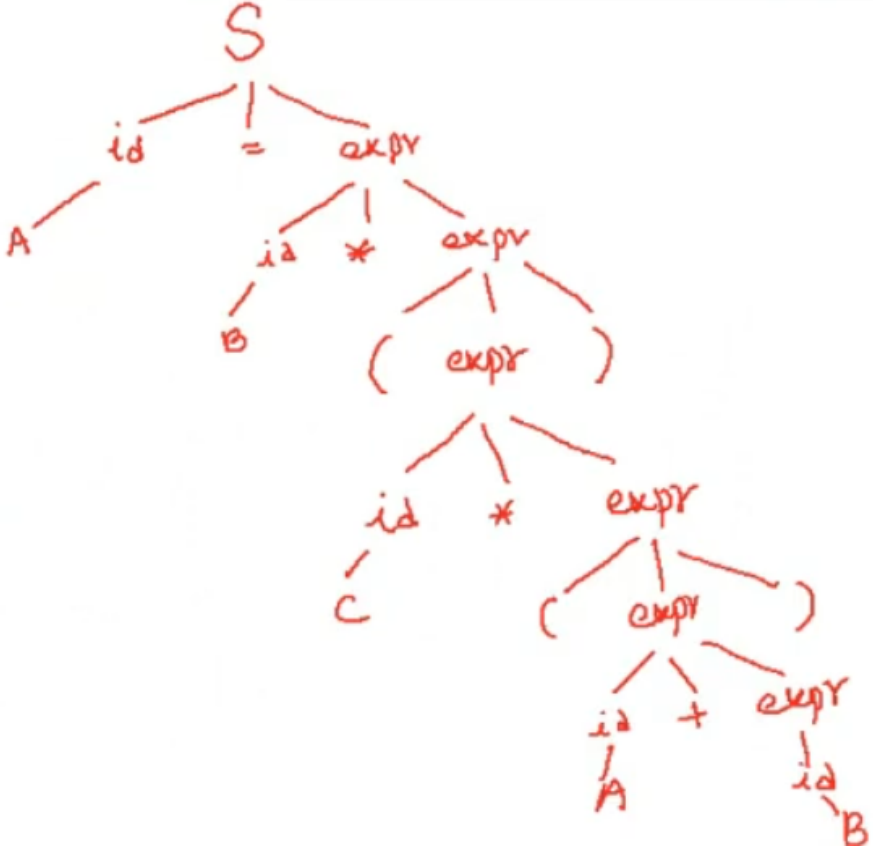
Or
<expression>→ <var> [(+|-) <var>]

Using the below Grammar- draw the parse tree for A=B * (C *(A+B))

A Grammar for Simple Assignment Statements

<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <id> + <expr>
| <id> * <expr>
| (<expr>)
| <id>

Exercise-4 : Solution.

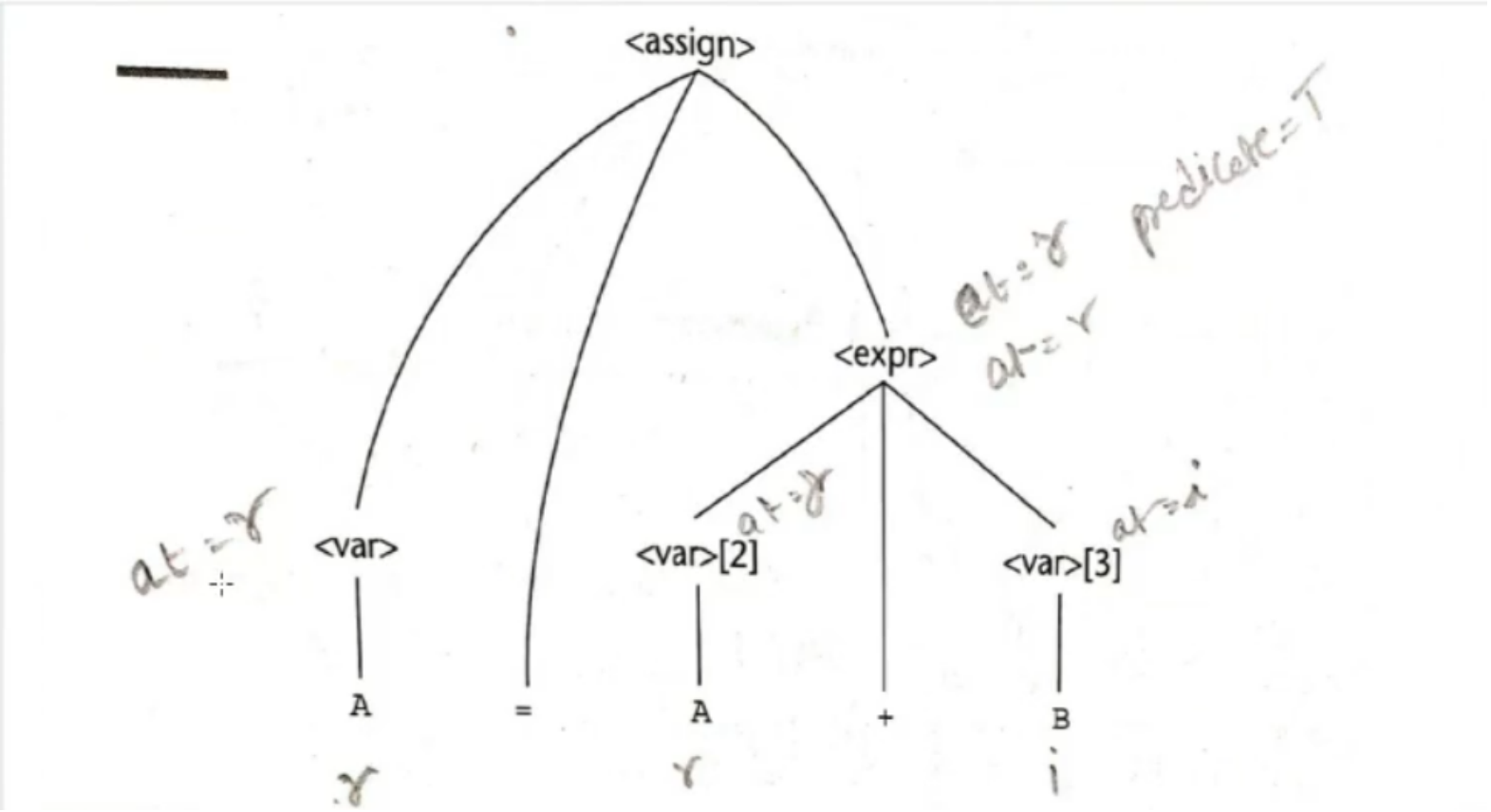


The syntax portion of our example attribute grammar is

<assign> → <var> = <expr>
<expr> → <var> + <var>
| <var>
<var> → A | B | C

An Attribute Grammar for Simple Assignment Statements

1. Syntax rule: <assign> → <var> = <expr>
   Semantic rule: <expr>.expected_type ← <var>.actual_type
2. Syntax rule: <expr> → <var>[2] + <var>[3]
   Semantic rule: <expr>.actual_type ←
   if (<var>[2].actual_type = int) and
   (<var>[3].actual_type = int)
   then int
   else real
   end if
   Predicate: <expr>.actual_type == <expr>.expected_type
3. Syntax rule: <expr> → <var>
   Semantic rule: <expr>.actual_type ← <var>.actual_type
   Predicate: <expr>.actual_type == <expr>.expected_type
4. Syntax rule: <var> → A | B | C
   Semantic rule: <var>.actual_type ← look-up (<var>.string)



Exercise – 5: See the following Grammar for assignment operation.

The syntax portion of our example attribute grammar is

<assign> → <var> = <expr>
<expr> → <var> + <var>
| <var>
<var> → A | B | C

Now give Attribute Grammar on the basis of example 6 in the Ch.3 of the text book, With following language rules.

1. Data types cannot be mixed in an expression.
2. But the RHS must match with LHS of assignment

An Attribute Grammar for the given Assignment Statements

1. Syntax rule: <assign> → <var> = <expr>
   Semantic rule: <expr>.expected_type ← <var>.actual_type

2. Syntax rule: <expr> → <var>[2] + <var>[3]
   Semantic rule: <expr>.actual_type ←
   if (<var>[2].actual_type = int) and
   (<var>[3].actual_type = int)
   then int
   else if (<var>[2].actual_type = real) and
   (<var>[3].actual_type = real)
   then real
   else mixed
   end if

   Predicate: <expr>.actual_type == <expr>.expected_type
   <var>[3].actual_type == <var>[2].actual_type

3. Syntax rule: <expr> → <var>
   Semantic rule: <expr>.actual_type ← <var>.actual_type
   Predicate: <expr>.actual_type == <expr>.expected_type

4. Syntax rule: <var> → A | B | C

Exercise – 6

Construct CFG for the following languages.

Language over ∑= {a, b} and any string contains exactly two ***a***s.

MOHAMMAD SAJJAD SHAIK 11:45 AM

S -> BaBaB
B -> bB | b | e

ANIRUDH A 11:47 AM
@Sajjad Wont B-> bB|e be enough?

MOHAMMAD SAJJAD SHAIK 11:47 AM
oh you're right

my bad