Virtual University	CS508-MODERN PROGRAMMING LANGUAGES (SUBJECTIVE) FROM MIDTERM PAPERS LECTURE (1-22)	Al Junaid Tech®
ajtechinstitute24@gmail.com	For More Visit: vujunaid.com	JUNAID MALIK (0304-1659294)

1) LISP stands for LIST processing language. It was designed by John McCarthy in 1959 as part of AI research at MIT to deal with problems of linguistic, psychology, and mathematics.

ANSWER:

LISP, an acronym for list processing, is a programming language that was designed for easy manipulation of data strings. Developed in 1959 by John McCarthy, it is a commonly used language for artificial intelligence (AI) programming. It is one of the oldest programming languages still in relatively wide use.In LISP, all computation is expressed as a function of at least one object. Objects can be other functions, data items (such as constants or variables), or data structures. LISP's ability to compute with symbolic expressions rather than numbers makes it convenient for AI applications.

2) ALGOL 58 – 1958 – Search for a "Universal Language"
ALGOL stands for Algorithmic Language. It was designed in 1958.

Salient features of the language are:
• Concept of type was formalized
• Names could have any length
• Arrays could have any number of subscripts
• Lower bound of an array could be defined
• Parameters were separated by mode (in & out)
• Subscripts were placed in brackets
• Compound statements were introduced (begin ... end)
• Semicolon as a statement separator was used
• Assignment operator was :=
• if had an else-if clause

Algol 60 - 1960
ALGOL 60 was a modified version of ALGOL 58. It added new features to the language. These included block structure (local scope), two parameter passing methods – value and name, subprogram recursion, and stack-dynamic arrays – run time size definition and space allocation. It did not have built-in I/O facility.

COBOL - 1960 COBOL was designed in 1960 to support business-oriented computation which requires fixed point arithmetic. It was designed to look like simple English to broaden the base of

computer users. It was thus required to be easy to use, even if that means it will be less powerful. Another design consideration was that it must not be biased by current compiler. It encountered problems because the design committee members were all from computer manufacturers and DoD branches and there were fights among manufacturers. It was the first language to add macro facility in a high-level language. It also introduced hierarchical data structures (records) and nested selection statements. The language supported long variable names (up to 30 characters) with hyphens, data division, and fixed-point arithmetic. It was the first language required by DoD and would probably have failed without support from DoD. It is still a very widely used business applications language and is very popular in business and government, much less at universities.

Question: Describe the following statement by SNONOBOL
:S(label2) F(label3)>> marks 2

Answer:

(label)

This is a command for unconditional jump to the label specified inside parentheses.

:S(label)

This is a command for jump. It is used to the specified label if the statement in the body was executed successfully or resulted in true Boolean value.

:F(label)

It is opposite to the second (label) :S (label). It the control jumps to the label if the statement is not executed successfully or resulted in false.

:S(label1) F(label2)

It is a combination of the second (label) :S(label) and the third :F(label). It states that go to label1 in case of success and label2 in case of failure.

Question: write two point that differentiate the LISP language from other language. Marks 2

Answer:

1. A LISP program has two types of elements:
   - Atoms
   - lists
2. The first symbol is the name of the function. The rest are its arguments.
   For example:
   >(+ 1 3 5)
   9

Question: Why we need single language Ada. Marks 2

Answer: The necessity for a single standard language was felt in 1975. The draft requirements were given the code name strawman. Strawman was refined to Woodman and then Tinman in 1976. Initial designs were submitted in 1978 and red and green short listed on the basis of these designs. Standard

requirements were then refined to steelman. The designs were refined additional and finally in 1979 Green was selected. DoD announced that the language will be called Ada. In 1995 revision of Ada (Ada 95) was developed by a small team managed by Tucker Taft. In both cases, the design need a public comment period where the designers answer back to public comments.

Question: Set the value of variable a HEIGHT =12 and Weight=5. Marks3

Answer:

>(setf (get 'a 'height) 12) ;

>(setf (get „a „weight) 5) ;

Question: What is difference b/w these two statements >(set-intersection L1 L2) and >(set-difference L1 L2) Marks 5

Answer:

>(union L1 L2) ; returns the union of the two lists

>(intersection L1 L2) ; returns the intersection of the two lists

>(set-difference L1 L2) ; returns the difference of the two lists

Question: 2 which data structure is more widely us in LISP programming?

Answer: Atoms and Lists.

Question: 4 Set the value of variable a HEIGHT =12 and Weight=5.

Answer:

(setf (get „a „height)12);
(setf (get „a „Width)5)

Question: 1; which predicate function will help for testing atoms?

Answer:

>(**member** 'b L)

Question: what is the functionality of KEYWORD INT|CAHR IN snobol4 2

Answer:

KEYWORD = 'INT' | 'CHAR'

This statement assigns the pattern „INT" | „CHAR" to the variable KEYWORD.

Question: 2.what is error in it >(sqrt x) in LISP 3

Answer: X is unbound because this symbol doesn"t have any value we will have to assign a value using setq, setf and set.

>(setq x 3) = 3

ÂL-JÛÑÂÎD TÊCH

Question: 3. how many element in it (a (d e)) related to lisp 2

Answer: There are two elements.

1. Atom = a
2. List = (d e)

Question: what is valid or invalid statement in ADA language 5
a=(1...5) integer
b=(1...5) integer
c,d=(1...5) float
condition
a=b

a=c
b=a
Answer:

a : integer (1..5); , //syntax is valid

a=c // condition is invalid

Question: What are the additional features added in COBOL language that were not included in SONOBOL. Marks: 2

Answer:

1. COBOL has introduced hierarchical data structures (records) and nested selection statements.
2. The COBOL language supported long variable names (up to 30 characters) with hyphens, data division, and fixed-point arithmetic.
3. These all feature in COBOL language are those which were added or enhanced and were not included in SNOBOL..

Question: Which programming language is considered to be the defacto standard for Artificial Intelligence community? Marks: 2

Answer:LISP was one of the earliest programming language

Question: Enlist any three major concepts that were introduced for the first time in LISP programming language. Marks: 3

Answer: From a programming language design point of view, LISP was the first to introduce the Following concepts:

• Conditionals: such as if-then-else construct.

• Function types: where functions are just like integers and strings

• Recursion: first language to support it.

• Dynamic typing: all variable are pointers.

• Garbage-Collection.
• Programs composed of expressions.
• A symbol type.
• The whole program is a mathematical function

Question: Name the famous two version of LISP programming:

Answer:

1) Schema 2) Common Lisp

Question: Why Ada does not use += and -= type of operators?

Answer: Such operators reduce readability. So Ada has not included operators like +=, -=, *=, /=, |=

Question No:

ÂL-JÛÑÂÎD TÊCH

Pattern matching is a powerful tool for syntax analysis in **SNOBOL**. Consider the following piece of code and identify if the pattern matching comes out to be success or failure justify your answer in either case.

FIRST = „SUMMER‟

SECOND = „SUM‟

FIRST SECOND

Answer:

If „SUM‟ will be searched in „SUMMER‟. Since it is present in it SUM („MER‟), it will result in success.

Question: Difference the car,cdr first and rest LISP function.

Answer:

first (or car) takes a list as an argument and returns the first element of that list.

For example:

>(first '(a s d f))

a

rest (or cdr) takes a list as its argument and returns a new list after removing the first element from the list.

For example:

>(rest '(a s d f))

(s d f)

Question: How many types of function are supported in SNOBOL 4?

Answer: SNOBOL 4 supports two types of functions:

1. built-in function which are known as primitive functions
2. user defined functions

Question: Write the name of the operator which are avoidable in Ada but not in C/C++?

Answer:

PlusEquals	+=
SubtractEquals	-=
MultiplyEquals	*=
DivisionEquals	/=
OrEquals	|=
AndEquals	&=

Question: Write the small program in Ada print the BS-computer-sciences.?

Answer:

```
With Ada.IO_txt
Use Ada.IO_txt
Begin
Procedure name is
New_line;
Put(“BS-computer-sciences”);
End name;
```

Question: Identify the list selector function used in LISP?

Answer: **List Selectors**

In order to select elements from a list, selectors functions are used. There are two basic selector functions known as first (or car) and rest (or cdr).

1. First (or car) is a list selector. It takes list as argument and return first element of list.
2. Rest(or cdr) is a list selector. It takes list as argument and return whole list after excluding first element.

Question: While Programming in SNOBOL4 which of the following operators must not have space between operator and operand? 2
a) Unary operators
b) Binary operators

Answer: Unary operators

Question: Briefly explain the data structures that are used in Ada for: 5
a) Encapsulation
b) Define extensible types

Answer:

Encapsulation: The primary structure used for encapsulation in Ada is called a package. Packages are used to group data and subprograms. Packages can be hierarchical, allowing a structured and extensible relationship for data and code. All the standard Ada libraries are defined within packages.

Extensible Types: Tagged types are usedto define extensible types.

Tagged Type: A tagged type is like a record which can beused to declare objects. Following is an example of a tagged type:

typePerson is tagged record

Name : String(1..20); Age : Natural; end record;

Question: Does Ada facilitate programmers to do programming as a human activity? If yes, give reason. Marks: 3

Answer: yes, ada believe in human programming .Programming as a human activity is the main goal of ADA. In ADA the whole philosophy is that a programmer is a human and all humans make mistakes. So to make it safer, Ada uses discriminants in a record that tell what type of data is currently stored there.

Question: Write a procedure in LISP programming language that takes the following two linked lists as arguments and returns the concatenation of these two lists by adding one list in front of another. Marks: 5

List a=(3 4 5)

List b=(C d 6)

Answer:

append takes two lists as arguments and appends one list in front of the other as shown below:

>(**append** '(a b) '(c d))

(A B C D) ; appends one list in front of another

Question: Write a function in LISP that return the difference of the two lists L3 and L4.

Answer:

These are set operations in LISP.

>(union L1 L2) ; returns the union of the two lists

>(intersection L1 L2) ; returns the intersection of the two lists

>(set-difference L1 L2) ; returns the difference of the two lists

Question: What was the main emphasis of Ada language? >> marks 2

Answer: Ada is a computer programming language originally designed to support the construction of long-lived, highly reliable software systems. Its design emphasizes readability, avoids Error-prone notation, encourages reuse and team coordination, and it is designed to be efficiently implementable.

Question: write the simple Ada program whether a number is event or not?

Answer:

```
declare
   N : Integer := 5;
begin
   if N rem 2 = 0 then
      Put_Line ("Even number");
   elseif N rem 2 /= 0 then
      Put_Line ("Odd number");
   else
      Put_Line ("Something went really wrong!");
   end if;
end;
```

Question: 3 write the ada program of given arguments Use for loop and use Put() method execpt PUt_Line() and write NEW_Line for new line the arguments are given below

```
*
**
***
****
*****
```

Answer:

```
procedure Loopprint is
Number_Of_Time: Integer:=5;
begin
for I in 1..Number_Of_Time loop
for J in 1..I loop
Put ('*'); end loop;
New_Line;
End loop;
end Nested_Loops;
end Loopprint;
```

Question: Suppose you want to remove a certain property attribute named „**WEIGHT**' from the following list symbols in LISP:
>(symbol-plist 'a)
(WEIGHT 20 HEIGHT 8)

Answer:
We can remove a property by using the **remprop** function as shown below:

```
> (remprop „a „WEIGHT)
T
>(symbol-plist „a)
(HEIGHT 8)
> (remprop „a „HEIGHT)
T
>(symbol-plist „a)
NIL
```

Question: Why Ada is considered to be a strongly typed language? Briefly explain.

Answer: ADA is strongly typed. A programming language is strongly typed if type errors are always detected and in ADA Errors can be signaled as exceptions and handled explicitly. Many serious errors such as computational overflow and invalid array indexes are automatically caught and handled through this exception mechanism, improving program reliability..

Question: Pattern Matching and string concatenation (Space). (SNOBOL)

Answer: In SNOBOL Space is used as the pattern matching operator and the statement has the following form: subject pattern There is no assignment operator in this case. Both the subject and the pattern are strings and this statement tries to match the pattern in the subject. It will be successful if the match is found and will result in failure otherwise. Example: TRADE = „PROGRAMMER" PART = „GRAM" TRADE PART

ÂL-JÛÑÂÎD TÊCH

Question: Write down the working and output of the string replacement function in SNOBOL with respect to the following code: 5

TEXT = "A {I,J} = A{A,B}+5"
OUTPUT = REPLACEMENT (TEXT, '{}', '[]')

Answer:

Replacement statement is used in conjunction with the pattern matching statement in the following manner.
subject pattern = object
In this case the pattern is searched in the subject and if found it is replaced by the object as demonstrated by the following example:
SENTENCE = „THIS IS YOUR PEN"
SENTENCE „YOUR" = „MY"
Since „YOUR" is present in the subject, it will be replaced by „MY", resulting in changing the value of SENTENCE to „THIS IS MY PEN".
If we now have the following statement
SENTENCE „MY" =
then SENTENCE will be further modified to „THIS IS PEN" as we are now replacing „MY" with a NULL string, effectively deleting „MY" from the subject.

Question: Does small alphabets and capital alphabets are consider same in Ada? Justify your answer? 2

Answer: No not case sensitive

Question: Enlist any three mathematical functions used in LISP. 3

Answer: There are many built-in function is LISP. This includes math functions as well as functions for manipulating lists. The math functions include:
+, -, *, /, exp, expt, log, sqrt, sin, cos, tan, max, min

Question:Why predicate is called a special function in LISP?

Answer:A predicate is a special function which **returns NIL** if the predicate is false, T or anything other than NIL, otherwise. Predicates are used to build Boolean expressions in the logical statements.

Comparative operators in LISP used as functions for numerical values and return a T or NIL. =, >, <, >=, <=;

For example 1:

(= (+ 2 4) (* 2 3))

T

Example 2:

(> (- 5 2) (+ 3 1))

NIL

Question: 3; concatenation n pattern matching in SNOBOL?

Answer:Pattern matching and manipulation is another very important feature of SNOBOL.

The first statement in this regards is the Pattern Matching Statement. Once again Space is used as the pattern matching operator. There is no assignment operator in this case. Both the subject and the pattern are strings and this statement tries to match the pattern in the subject. It will be successful if the match is found and will result in failure otherwise. This is confirmed with the help of the following example:
TRADE = „PROGRAMMER‟
PART = „GRAM‟
TRADE PART

Question: when Put_Line method is called in Ada does need to calL New_Line method or not?

Answer: No need because it already add new line. When use only put() then use new line.

Question: dotime dolist men sy function tha koi , >> marks 3

Answer: In lisp are two basic statements for that purpose: **dotimes** and **dolist.**
dotimes
dotimes is like a counter-control for loop. Its syntax is given as below:
(**dotimes** (count n result) body)

It executes the **body** of the loop n times where count starts with 0, ends with n-1.

dolist
The second looping structure is **dolist.** It is used to iterate over the list elements, one at a time. Its syntax is given below:

(**dolist** (x L result) body)

Question: 2; Arrays in SNOBOL?

Answer: pg 42

Question: what is tagged in Ada?

Answer:**Tagged Type**
A tagged type is like a record which can be used to declare objects.
Following is an example of a tagged type:
type Person **is tagged record**
Name : String(1..20);
Age : Natural;
end record;

Question: How many types of statements are used for pattern building in SNOBOL. 2

Answer:

There are two type of statements for pattern building. These are Alternation and Concatenation.

Question: What do you know about the increment/ decrement or predecessors and successor in enum types in Ada explain with suitable example. Marks = 3

Answer:

An enumeration type in Ada is a sequence of ordered enumeration literals just like C:

Type Colors is(Red, Indigo, Orange, Green, Yellow, Blue, Violet);

Type State is (Off, Powering_Up, On);

It is however different from C in many respects:

There is no arithmetic defined for these types.

For example:

S1, S2 : State;

S1 := S1 + S2; -- Illegal

One can however add/subtract one (sort of increment and decrement) using the Pred and Succ as shown below:

State‟Pred (S1)

State‟Succ (S2)

Questions: Which is the largest language in which most of the programming done today?

Answer:

Ada – 1983 – History‟s largest design effort It involved a huge design effort, involving hundreds of people, much money, and about eight years. It introduced Packages - support for data abstraction, elaborate exception handling, generic program units, and concurrency through the tasking model.

Question: Explain any unique thing related with lisp conditional control statements. 5

Answer: Conditional control:

If when and Cond LISP has multiple conditional control statements. The set includes if, when, and cond. In the following pages we study these statements one by one.

If statement

The if statement has the following syntax:

(if <test><then><else>)

That is, if statement has three parts: the test, the then part, and the else part. It works almost exactly like the if statement in C++. If the test is TRUE then the then part will be executed otherwise the else part will be executed. If there is no else part then if the test is not true then the if statement will simply return NIL.

Here is an example that shows the if statement:

> (setq SCORE 78)

> 78

> (if (> score 85) „HIGH

(if (and (< score 84)(> score 65)) „MEDIUM „LOW))

> MEDIUM In the above if statement, the thenpart contains „HIGH and the elsepart is another if statement. So, with the help of nested ifstatements we can develop code with multiple branches.

Question: write a two dimensional array of siz (2x8) in sonobl and intialize it to zero.Marks 3

Answer:

The following statement creates a two dimensional array of 2 x 8 and each cell is initialized to NULL string.

N = ARRAY(„2,8‟)

Q2 Lists are used in lisp write top elements of the list given as (add 3(multp 3 4)) and (job car school) Marks 3

Answer:

The first-level elements in LISP are called top-level elements. For example top elements of list (a b c) are a, b, and c. Similarly, top elements of list (a (b c)) are a and (b c). An empty list is represented by nil. It is the same as ()

Question .tell the out put if

defun y-min (x)(- x,y)

set y 2

y-min 20

Answer: -18

Question: Overloading in ada and c++?

Answer: Operator Overloading in ADA

Ada allows a limited overloading of operators. The exception in Ada is that the assignment operator (:=) cannot be overridden. It can be overridden in case of inheritance from a special kind of "abstract class". When you override the equality operator (=) you also implicitly override the inequality operator (/=).

QUESTION: Explain briefly the expression of the LISP language.

Answer: - **S-expression**

An S-expression(S stands for symbolic) is a convention for representing data or an expression in a LISP program in a text form. It is characterized by the extensive use of prefix notation with explicit use of brackets (affectionately known as Cambridge Polish notation).

S-expressions are used for both code and data in Lisp. S-expressions were originally intended only as machine representations of human-readable representation of symbols, but Lisp programmers soon started using S-expressions as the default notation.

S-expressions can either be single objects or atoms such as numbers, or lists.

Question: what is the role of atom in LISP? 17-12-2016 current

Answer: - Atoms:

Atoms include numbers, symbols, and strings. It supports both real numbers and integers.

Question: write array in ADA using Days of a week as indexes,

just like example given on page: 51

Answer:

For example if we define Days as below:

type Days is (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday); Then we can create an array which uses Days (or any sub-range of Days) as the array indexes as shown below.

type Daily_Sales is array(Days) ofFloat;

Now the type Daily_Sales is an array type of size 7 and indexes Monday through Sunday. Also note that these literals will be used as indexes in the statements that reference array elements.

Question: How Bitwise Exclusive Or, and Not is represented in Ada?

Answer:-

Bitwise Exclusive Or	=Xor
Bitwise Not	=Not
Logical Not	=Not

Question: Identify Local and Global variable

>(defun y-plus(x)(+x y))

Answer:-

This function adds x to y where x is the parameter passed to the function and y is a global variable since it is not defined inside the function. It works in the following manner:

>(setq y 2)

>(y-plus 23)

25

With this we introduce the concept of local and global variables. Local variables are defined in function body. Everything else is global.

Question: Whether polymorphism is allowed in Ada or not? Briefly state. (2 marks)

Answer: Ada 83 supported a form of polymorphism through the use of overloading that allowed the developer to reuse a procedural name or operator by redefining the context and semantics. For example, when using integer numbers, the + sign adds one integer to another. The + operator could be redefined to work with an abstract data type consisting of a set of colors and provide the proper color mix (i.e., red + yellow = orange). This polymorphic behavior had to be resolved at compile time. Ada 95 supports a true polymorphic operation through the use of class-wide types and type extensions, which can contextually recognize the appropriate operation to perform.

Question: Is there anything like templates as in C++.

Answer:

Generics are like templates in C++ and allow parameterization of subprograms and packages with parameters which can be types and subprograms as well as values and objects.

12-17-2016

Which language had a strong influence on one of the high level system programming language like C?

Answer: ALGOL 68 was from the continued development of ALGOL 60, but it is not a superset of that language. Its design is based on the concept of orthogonality.

ALGOL 68 had even less usage than ALGOL 60 but had strong influence on subsequent languages, especially C, Ada and Pascal.

ÂL-JÛÑÂÎD TÊCH

Question: Which of the languages first provided a structured exception handling mechanism among Ada and C++?

Answer: PL/1 has introduce the concept of exception handling. But ADA has elaborated this concept.

- Ada was the first language to provide a structured exception handling mechanism.
- Ada includes an exception handling mechanism so that these problems can be dealt with at run time.
- Errors can be signaled as exceptions and handled explicitly.

Question: Considering „Generalization" as an important factor that helps to manage complexity of any programming language. Briefly mention how could generality be considered as an important language evaluation criteria?

Answer:

Generality is also an important factor that helps to manage complexity of any programming language and deals with the applicability of the language to a range of different domains.

For example:

C is more general purpose than FORTRAN or LISP. So, therefore can be used in more domains than these two languages.

Question: Explain the concept of dangling else with an example. And also explain the solution to this problem.

Answer:

"Dangling else" in C explains the association of single else statement in a nested if statement.

In nested if statements, when single "else clause" occurs, the situation happens to be dangling else.

For example:

```
if (condition)
if (condition)
if (condition)
else
printf("dangling else!\n");  /* dangling else, as to which if statement, else clause associates */
```

Solution to the problem:

In such situations, else clause belongs to the closest if-statement which is incomplete. That is the innermost if-statement. We can make else clause belong to wanted if statement by enclosing all if statements in block outer {} to which if statement to associate the else clause.

For example:

```
if (condition)
{
if (condition)
if (condition)
} else
 printf("else associates with the outermost if statement!\n");
```

ÂL-JÛÑÂÎD TÊCH

Question: Suppose you have to write a program in Ada Language for calculating marks of a student for 3 subjects. The total marks for each subject is as follow:

Subject1= 100

Subject2= 200

Subject2= 400

How will you restrict users to not enter values less than 0 and greater than total marks? Also write code snippet for this.

Note: The use of loops and conditions are not allowed.

Answer:

```
with Ada.Text_IO;
use Ada.Text_IO;
procedure Subject is
S1: String(1 .. 100)
S2: String(1 .. 200)
S3: String(1 .. 400)
Last: Integer;
begin
Put_Line("Subject1:");
Get_Line(S1, Last);
Put_Line("Subject2:");
Get_Line(S2, Last);
Put_Line("Subject3:");
Get_Line(S3, Last);
end Subject;
```


CURRENT

1. **How we can increase readability by using different languages, write 5 at least. (3 marks)**

   Answer:

- Simplifying sentence structure: Using shorter, simpler sentences can improve readability for readers of all language backgrounds.
- Using active voice: Writing in active voice (as opposed to passive voice) can make sentences clearer and more direct.
- Using common words: Avoiding technical jargon or uncommon words can make text more accessible to a wider audience.
- Using headings and subheadings: Breaking text up into smaller sections with headings and subheadings can make it easier to skim and understand.

- Using images and other media: Adding images, videos, or other media can help to clarify or illustrate information, making it easier to understand.

2. **Type a record name school by using Ada language**
   ANSWER:

```
type School is record
   Name : String;
   Address : String;
   Phone_Number : String;
   Principal_Name : String;
   Number_of_Students : Integer;
end record;
```

3. **Suppose you want to remove a certain property attribute named „WEIGHT' from the following**
   list symbols in LISP:
   >(symbol-plist ‘a)
   (WEIGHT 20 HEIGHT 8)
   Answer:

You can use the function remprop to remove a property from a symbol's plist. The syntax is: (remprop symbol property).

So, to remove the "WEIGHT" property from the symbol 'a', you would use the following code:

```
(remprop 'a 'WEIGHT)
```

After running this code, calling (symbol-plist 'a) should return (HEIGHT 8)

4. **Difference between functions and processers of Ada?**
   Answer:

In Ada, a function is a subprogram that returns a value, while a procedure is a subprogram that does not return a value.

Functions are used to perform calculations or transformations on input data and return the result, while procedures are used to perform actions, such as printing output or modifying the state of a program.

Functions have return types while procedures do not.

Functions can be used in expressions and can be passed as arguments to other subprograms, while procedures cannot. Procedures can only be called.

For example, you might use a function to calculate the square of a number, and a procedure to print the result to the screen.

```
function square(x: Integer) return Integer is
begin
  return x * x;
end square;

procedure print_result is
  result : Integer;
begin
  result := square(4);
  Put_Line("The square of 4 is " & Integer'Image(result));
 end print_result;
```

5. **parent class is employee and child class is teacher write the code and explain and second long question was given 7 taske of gui and perform him like as creat two button creat connection and add button set fram size and visible and gui method call in class and make object etc**
   Answer:

```
with Ada.Text_IO;  use Ada.Text_IO;

type Employee is tagged null record;
  procedure Print_Name(Self : in out Employee);

type Teacher is new Employee with null record;
  procedure Print_Subject(Self : in out Teacher);

procedure Employee.Print_Name(Self : in out Employee) is
begin
  Put_Line("Name: John Doe");
end Print_Name;

procedure Teacher.Print_Subject(Self : in out Teacher) is
begin
  Put_Line("Subject: Math");
 end Print_Subject;
```

6. **Write three descendents of c languageAnswer:**

Answer:

- C++: C++ is an extension of C, which was developed by Bjarne Stroustrup in 1983. C++ introduced the concept of classes and objects, which allows for the creation of user-defined data types and the use of object-oriented programming techniques. It also includes features such as function overloading, templates, and exception handling.
- C#: C# is a modern, object-oriented programming language developed by Microsoft in 2000. It is designed for the Microsoft .NET framework and is commonly used for developing Windows desktop and web applications. C# is similar to C++ and Java in terms of syntax and features, but it also includes features such as garbage collection and built-in support for threading and parallel programming.
- Objective-C: Objective-C is an object-oriented programming language that is a descendant of C. It was developed in the early 1980s by Brad Cox and Tom Love. Objective-C is widely used on Apple's macOS and iOS operating systems, as well as their respective APIs, Cocoa and Cocoa Touch. Objective-C is known for its dynamic binding, which allows for the creation of runtime-modifiable programs.

These are just a few examples of the many descendants of C language, there are many more in the market like Go, Rust, Swift, etc.

7. **Write three built in functions used in lisp**

   Answer:

- car: The car function returns the first element of a list. For example, (car '(1 2 3)) would return 1.
- cdr: The cdr function returns the rest of the elements in a list after the first element. For example, (cdr '(1 2 3)) would return the list (2 3).
- cons: The cons function is used to construct a new list by adding an element to the front of an existing list. For example, (cons 1 '(2 3)) would return the list (1 2 3).
- These are just a few examples of the many built-in functions in Lisp, there are many more like eq, equal, atom, list, +, -, *, /, etc.

8. **Following is the snippet from SNOBOL 4. Calculate the value of x,y ,z where ** sign is used for exponential.**

1). x= 2*3*2

2). y= 3*2*3

3). z= x+y2**

Answer:

In SNOBOL 4, the ** operator is used for exponentiation. So, with that in mind:

- x = 2*3*2 = 12
- y = 3*2*3 = 18
- z = x + y**2 = 12 + 18**2 = 12 + 324 = 336

So, the final values of x, y, and z are 12, 18 and 336 respectively.

9. **Pascal main simple "Hello World" program likhna tha, with starting and ending statements as well.**

   Answer:

```
program HelloWorld;
begin
  writeln('Hello World');
  end.
```

10. **Programming language ki simplicity aur readability**

    Answer:

    - Simplicity refers to how easy it is to understand and use a language. A simple language is one that has a small set of features, is easy to learn, and has a straightforward syntax. Simple languages are often preferred for small projects, learning to code, or for people who are not professional programmers.
    - Readability refers to how easy it is to understand the code written in a particular language. A readable language is one that follows a consistent style, uses meaningful variable and function names, and has clear and concise code. Readable code is often preferred by professional programmers as it makes it easier to maintain and debug the code.

11. **write program "Hello World" in ADA**

    Answer:

```
with Ada.Text_IO; use Ada.Text_IO;

procedure Hello_World is
begin
   Put_Line("Hello World!");
  end Hello_World;
```