

Filling All Details in Part - I are Compulsory and to be filled by the Candidate with his / her own handwriting.

(PART - I)

1.) Examination (Class) :	<u>BE CSE</u>
2.) Subject:	<u>Principles of programming language</u>
3.) Paper:	<u>CS 504</u>
4.) Option:	<u>Regular</u>
5.) Medium:	<u>English</u>
6.) Date:	<u>27/2/22</u>
7.) Total No. of Sheets Used:	
8.) Mode of Submission of Answer sheet:	<u>Online</u>

Roll No. (in figures): 2019342

Roll No. (in words): Two one Nine
Three four two

Examination (Class): BE CSE

Subject: POPL

Paper: CS 504

Option: Regular

Subject Code: 6789

Exam Code: 0917

(PART - II) Evaluation: Candidates should not write in this section

CODE NO. (To be allotted by the Evaluator) _____ ID NO. _____

Q1. ____ Q2. ____ Q3. ____ Q4. ____ Q5. ____ Q6. ____ Q7. ____ Q8. ____

Q9. ____ Q10. ____ Q11. ____ Q12. ____ Q13. ____ Q14. ____ Q15. ____ Q16. ____

Total Marks in figures: _____

Name and Signature of the Evaluator

Total Marks in words : _____

START WRITING YOUR ANSWERS BELOW

Ans 7

a) When a variable is declared as being a pointer to type void it is known as generic pointer. Since we cannot have a variable of type void, the pointer will not point to any data & ∴ cannot be dereferenced. To use it, we have to cast it into another type of pointer.

void pointer is an approach towards generic functions & generic programming in C.

Syntax - void * ptr-name

Example - main()

```

    { int i;
      char c;
      void *data;
      i = 6;
    }
  
```

```

c = 'a';
data = &i;
printf("points to value %d", *(int*)data);
data = &c;
printf("points to character %c", *(char*)data);
return 0;
}

```

A Template is a generic class or other unit of source code that can be used as the basis for unique units of code. In C++, there are standard template libraries from which programmers can choose individual template classes to modify.

b) In a rule based language - programs are composed by a set of rules. Each rule has 2 parts, called head that represents consequence and body that represents premises. A rule based system attempts to derive execution instructions from a starting set of data & rules.

Object Oriented language is a high level computer language that implements objects & their associated procedures within the programming context to create software programs.

Object oriented language uses an object oriented programming technique that binds related data & functions into an object & encourages reuse of these objects within the same & other programs.

c) Unification is a process of making 2 different logical atomic expressions identical by finding a substitution. Unification depends on the substitution process.

It takes 2 literals as input & makes them identical using substitution.

Rules - 1) Predicate symbol must be same, atoms or expression with different predicate symbol can never be unified.

2) Number of arguments in both expressions must be identical.

3) Unification will fail if there are 2 similar variables present in same expression.

d) Type checking is the operation on which arguments that only can be applied for

→ Static type checking performs the type checking operation before the execution of the program. To perform this operation, the arguments, expressions, variables must be given a data type.

→ Dynamic type checking performs the type checking operation at the time of program execution. To perform this operation the arguments, expressions, variables must be given a data type.

e)

An activation record is a data structure that is activated / created when a procedure / function is invoked & it includes the following data about the function.

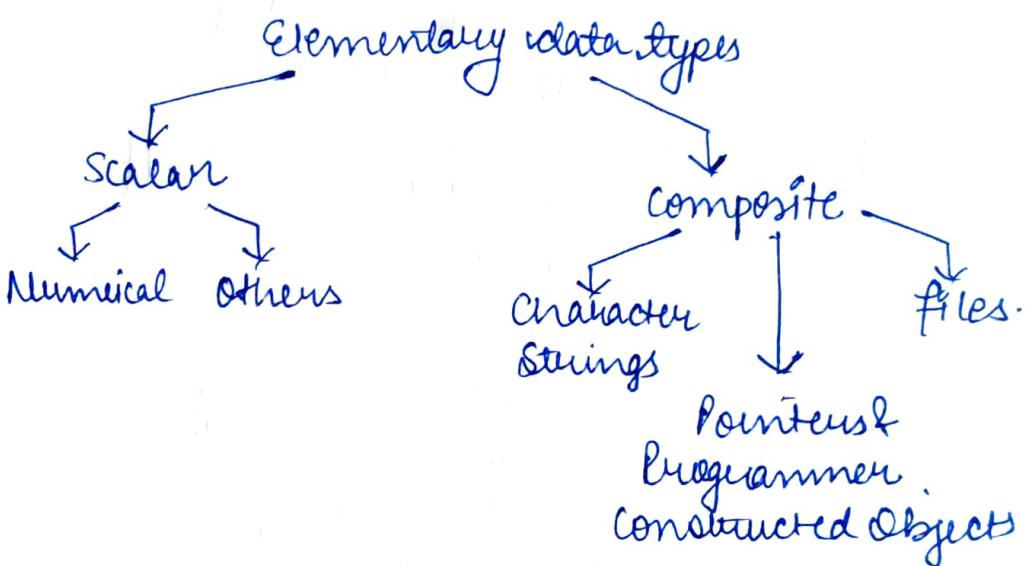
C language Record -

→	local data
	old SP
	Return Value
	ReturnAdd.
	No. of Args.
	ActualParams.

Unit II

Ans VI

a)



A composite data type or compound data type is any data type which can be constructed in a program using a programming language's primitive data type & other composite types.

It is sometimes called a structure or aggregate data type, although the latter term may also refer to arrays, lists, etc. The act of constructing a composite type is called a composition. Composite data types can also be contrasted with scalar variables.

- ↳ It stores a collection of individual values all under one variable name.
- ↳ and allows individual values to be accessed & for assignment & use.

1) C++ / C

Structure - `typedef struct {`

```
    char name [25];
    int id;
    char group;
    float marks [5]
    double interest;
```

`} student;`

`struct student1`

`cout << "Student1 ID" << student1.id;`

Union - `union Student {`

```
    char name [25];
    int id;
    char group;
    float marks [5];
    double interest;
} st1, st2;
```

3) Enumeration - enum weekdays;
enum weekend;

Python -

1) Array - list or []

names = ["A", "B", "C"]

2) Record - tuple or ()

tup = ("A1", "A2", "A3")

3) Union - dict or {}

dit = {

"Name": "A",
"Age": 4,
"Group": "B"

Java -

1) Arrays - int [] x = {0, 10, 20} ;

2) Classes - public class Point

{ int x;

int y;

Point (int un-x, int un-y)

{

x = un-x;

y = un-y;

void MovePoint (int dx, int dy)

{

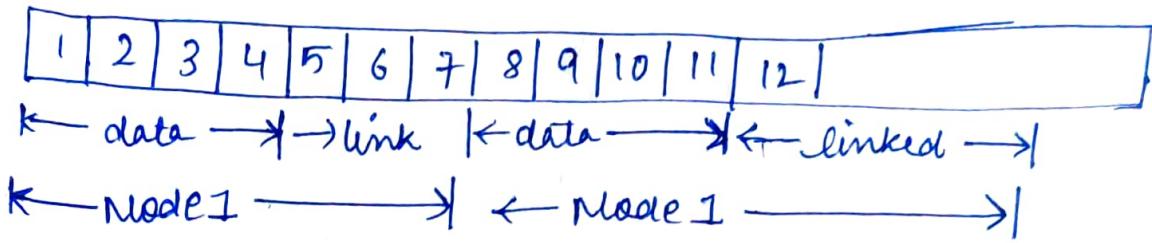
x += dx;

y += dy; }

(b) In deciding on a data representation for a given data object one must take into consideration the cost of performing different operations using that representation.

Strings are stored in 3 types of structures -

- 1) fixed length structure
- 2) variable length structure
- 3) linked structure.



The memory is divided into nodes of size 4 with a link field that is 2 characters long. Deletion of a substring can be carried out by replacing all characters in this substring by 0s or freeing nodes in which data fields consist of 0s only.

for example in C -

using pointers - `char *str = "ABC";`

Ans VII

Concurrency is when 2 or more sequences occur "in parallel".

Multiprogramming -

- 1) Single processor runs several programs at the same time
- 2) Each program proceeds sequentially.
- 3) Actions of one program may occur b/w 2 steps of another.

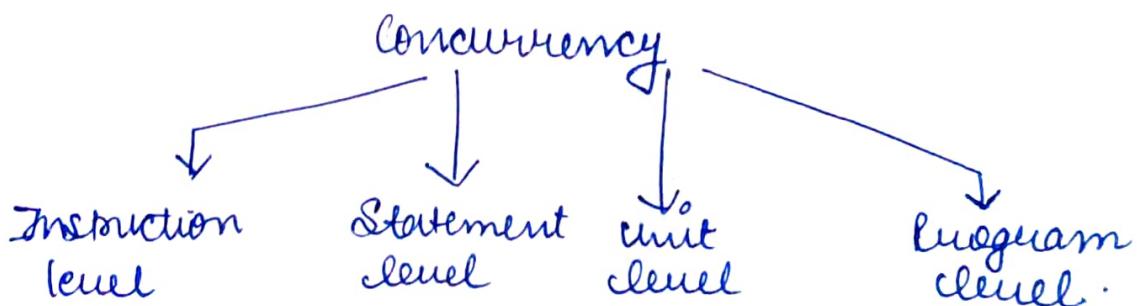
Multiprocessors -

- 1) Two or more processors
- 2) Programs on one processor communicate with programs on another.
- 3) Actions may happen simultaneously.

Eg - Browsing a webpage -

Multiple concurrent activities like

- ↳ Thread for each image load.
- ↳ Thread for text rendering.
- ↳ Thread for user input.



Explicit -

- 1) Create processes explicitly.
- 2) Explicitly interaction b/w processes.

Implicit -

- 1) Rely on compiler to identify potential parallelism.
- 2) Instruction-level & loop-level parallelism can be inferred, but inferring subroutine-level parallelism has less success.

The language support includes -

- 1) Communication
 - Synchronous
 - Asynchronous
- 2) Synchronization / Concurrency control
 - Synchronization primitives

Sometimes processes don't need to communicate explicit values to cooperate. They might have to synchronize activities.

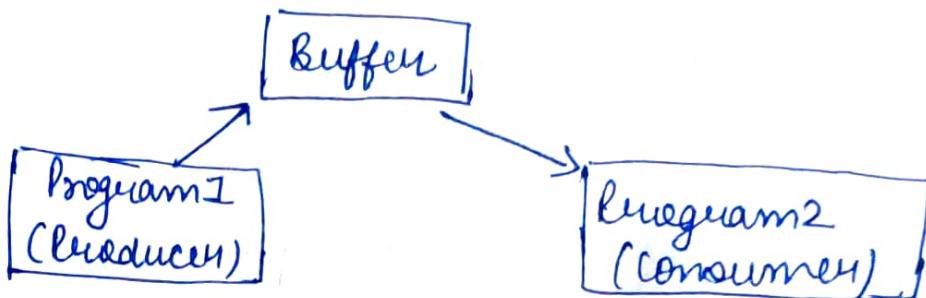
Synchronization is a mechanism that controls the order in which tasks execute.



Cooperation is required when Task A must wait for task B to complete some specific activity before Task A can continue its execution.

Competition Synchronization is required between 2 tasks when both require the use of some resource that cannot be simultaneously used.

Cooperation synchronization-



Program1 produces data & program2 uses the data.

Synchronization is needed as -

- 1) The consumer must not take the data if buffer is empty.
- 2) The producer unit cannot place new data in the buffer if it is not empty.

Competition synchronization-

We have 2 tasks (A & B) & a shared variable (TOTAL)

Task A must add 1 to total, Task B must multiply TOTAL by 2.

Each task accomplishes operation using -

- fetch the value of TOTAL
- Perform arithmetic operation
- Put the new value back in TOTAL.

Total has an original value of 3.

(1)

Without competition synchronization, 4 values could result from the execution of 2 tasks:

- If A completes before B begins \rightarrow 8
- If A & B fetch TOTAL before either puts the new value back in, then
 - if A puts new value, stack first \rightarrow 6
 - if B " " " " " " \rightarrow 4
- If B completes before A begins \rightarrow 7.

This condition is also named as a race condition as 2 or more tasks race racing to use shared resources & result depends on which one gets there first.

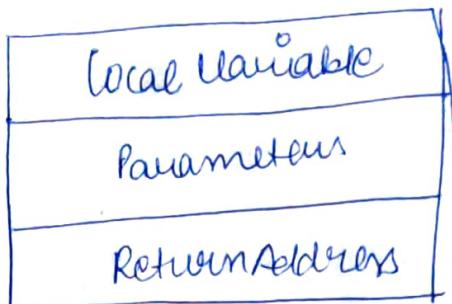
Unit I

Ans III An Activation Record is a private block of memory associated with an invocation of procedure. It is a runtime structure used to manage a procedure call.

An activation record is pushed into stack when a procedure is called & it is popped when control returns to caller function. The format or layout of the non code part of an executing subprogram is called an activation record.

Simple Subroutines are those that cannot be nested & all variables are static. A simple subroutine consists of 2 parts - code & data. Both parts have fixed sizes.

The form of an activation record is static.
The activation record instance is a concrete example of an activation record, corresponding to one execution.



Since the activation record instance of a simple subprogram has fixed size, it can be statically allocated.

The code & activation records have 4 program units

- 1) Main
- 2) A
- 3) B
- 4) C

Main calls A, B & C

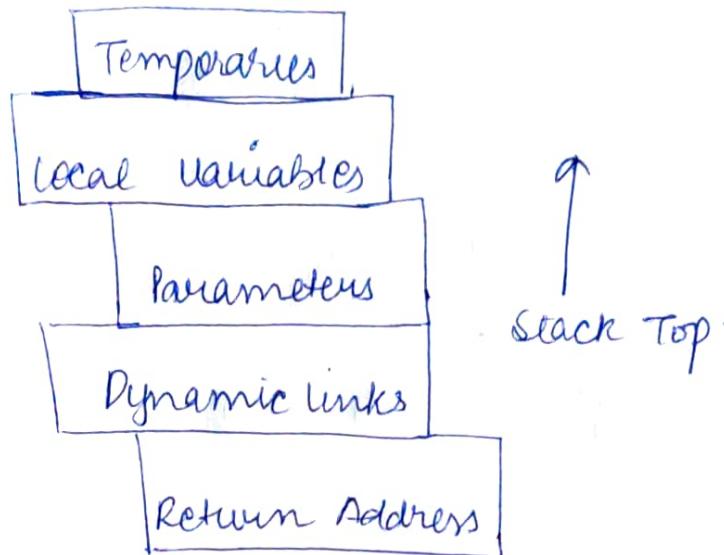
When each program is compiled, its machine code along with a list of references to external subprograms are written to a file.

The code is linked using a linker for Main to create an executable file. It finds & reads all references in the memory. It sets target addresses of calls to those subroutine entry addresses.

In the context of specification & implementation requires more complex activation records. The compiler must generate code to cause implicit allocation & deallocation of local variables.

(13)

Since the return address, dynamic link & parameters are placed in the activation record instance by the caller, these entries must appear first. Local variables are allocated possibly initialized in the caller so they appear last.



Dynamic links are used in destruction of current activation record instance, when procedure completes its execution.

The dynamic link is required in some cases because there are other allocations from stack such as temporaries.

e.g main()

```

Point f;
f = fact(3);
    
```

```

int fact(int n)
    
```

```

{ if (n == 1)
    return 1;
else
    
```

```

        return (n * fact(n - 1));
    } 
```

AR
for
fact

first cell	
Main	f
clocal	
Returnvalue	
Act. Parameter	3
Dynamic link	

AR
for
fact

(14)

Second & third cell	
Main	4
clocal	f
fact	4
returnvalue	6
parameter	3
dynamiclink	
fact	4
returnvalue	2
parameter	2
dynamiclink	
fact	4
returnvalue	1
parameter	1
dynamiclink	1

Ans IV

a) Resolution

1) It is Inference mechanism, when we resolve 2 clauses we get one new clause.

2) Eg - All women like shopping -
Olivia is a woman.

Ans - Olivia likes shopping.

Backtracking

1) Procedure in which prolog searches truth value of different predicates by checking whether they are correct or not.

Backtracking is quite common in algorithm designing & programming environments (15)

- 3) In Prolog, until it reaches proper destination
It tries to backtrack & stops when destination is reached.

Example - X & Y can play but a boy can play to a girl. So X = boy & Y = girl.

logic -

IP - play(X, Y)

X = tom

Y = alice?

X = tom

Y = lilly.

KB

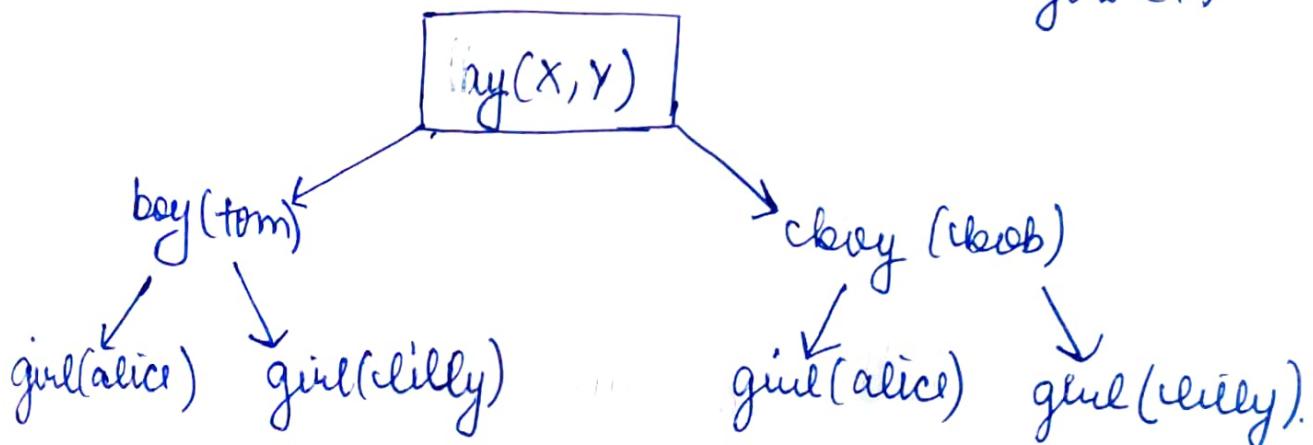
boy(tom),

boy(kob),

girl(alice),

girl(lilly)

play(X, Y) :- boy(X),
 girl(Y)



	Static	Dynamic
①	Variables get allocated permanently till the program executes or function call finishes	Variables get allocated only if the program until gets active.
②	Done before program execution	done after during program execution
③	Uses stack	Uses heap.
④	Less efficient	More efficient
⑤	No Memory Re-usability	Memory is re - usable.
⑥	Memory size cannot be reshaped	Memory size can be reshaped.
⑦	faster	slower.
→	Used for arrays	→ used for linked lists

Dangling Reference— It is a pointer to an element that has been returned to free space list means dangling pointer, so that it has been released to free. It occurs when object is deleted or deallocated without modifying the value of pointer so that the pointer still points to the memory location of the deallocated memory.

Garbage occurs when entire access path to a
data object is destroyed but the data object
continues to exist & the data object then said to
be garbage.

(7)

Garbage collection is the reclamation of chunks
of storage holding objects that can no longer
be accessed by a program.