24/07/23

Page No.
Date

Unit-1 - Principals of object Oriented Programming
- (weightage -14m)

Questions

1. Write any four applications of OOP. - 2m
2. State and describe characteristics of OOP - 2m.
3. State difference between OOP x POP - 2m/4m
4. Write any four benifits of OOP - 4m.
5. In Procedure Oriented Programming all data are shared by all functions. is the is this statement true? Justify your answers. - 4m
6. State difference between OOP and POP 2m

26/07/2023

4m

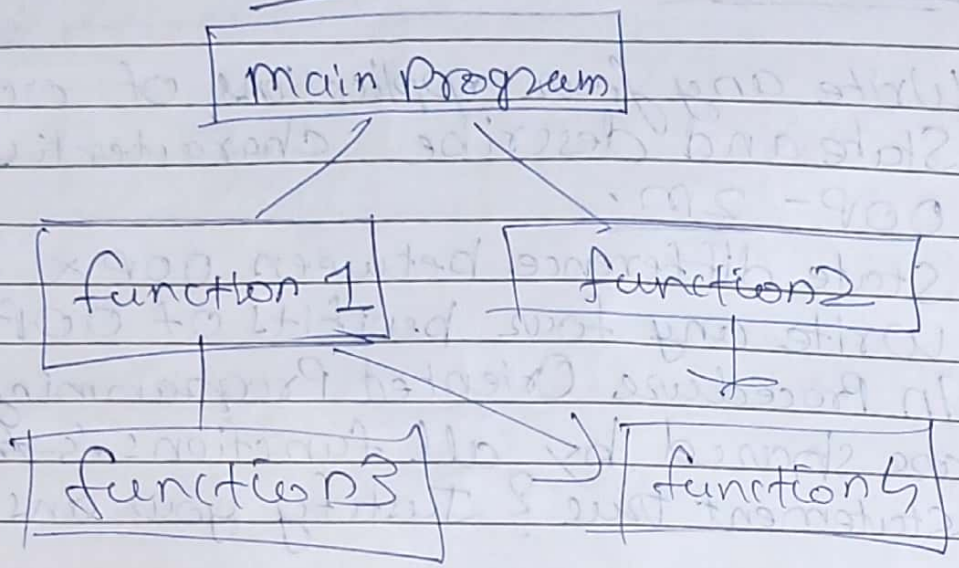
In Procedure Oriented Programming all data are shared by all functions is this statement true? Justify your answer.

⇒ POP and OOP both are programming approaches used for computer programming. These are nothing but style of writing program.

1. POP is traditional way of programming where an application problem is eview as sequence of state (algorithm).
2. A program in POP is sets of instructions here a programs are divided into different procedures. (Also known as functions) and each procedure contains set of instructions that performs specific tasks.
3. In POP each function has clearly definied purpose. A number of functions can be group together into larger entity called mod

④ Programs in POP contains of controlling function known as the main which controls execution of other functions.

⑤ POP



Program structure in POP.

⑥ When a call to the function is made the program control is passed to the function and all the instructions in that function are executed one after other.

⑦ After executing all the instructions the program controls returns to the functions from where the call is made.

⑧ ~~POP~~ In function in POP can have it's own data known as local data.

⑨ Multiple function can access same data known as global data.

⑩ In POP, many important data items are placed as global so that they can be ~~as~~ accessed by all the other functions.

⑪ So it true that in POP not all data but some important data are shared by important all the functions.

Page No.
Date

Characteristics and features of POP

1. In POP, the application programs are organized designed and written around the procedures.
2. More focus is lead on procedures that is nothing but doing things rather than data.
3. In POP, data moves openly around the system from functions to functions because it is made global.
4. POP language uses top down programming approach.

Top down approach

A bigger problem is divided into smaller module.

Bottom up approach

Smaller problems are solved and then they are integrated to find solution for bigger problems.

OOP - object oriented Programming.

— To overcome the limitations of POP, OOP has been developed.

— OOP not only includes the features of POP but also includes new powerful feature

— The main idea behind OOP languages is to encapsulate (merge together) both data and functions that operates on this data into a meaningful unit known as object

— In OOP, data is considered as critical element and it's tied to functions as a set of data that is encapsulated in an object known as member data and the functions which operates on this data known as member functions.

LM Q// Write any four benifits of OOP/Advantages.

① Reusability of the code is main advantage of OOP. Redundant (unnecessary) can be eliminated through inheritance feature of OOP.

② Code sharing: In OOP, if software team have develop some code we can directly use it in our program. No texting is required. Reusability is refer to using our own code while code sharing refers to code written by someone else.

③ Rapid prototyping (update versions): object orientation helps in rapid prototyping. When basic design is done. elementary (new) objects can be design with minimum of methods.

POP- focuss (fcn^n)
OOP- focus funct data
object.

Page No.
Date

(4) Information Hindi Hiding : In OOP there is lot of safety to the software. Here complier does not allow pc to make any error. In OOP, (5) the programs and data are more secured due to data hiding feature.
(5) In OOP program can easily be broken down into objects which are helpful to represent real world entities (object)
(6) New data-types can be easily created through data abstraction feature of OOP.
(7) The six pillers pillars of oop (class, object, inheritance, polymorphism, data encapsulation, data abstraction) make it easier to develop, test, maintain the code.
(8) In OOP importance is given to the data rather than procedures or functions because it works as a real world.

Q/ State and describe characterics/features of OOP.
Ans =>
- In OOP, the application programs are organised designed, and written around the data making object oriented programming data centric.
- Unlike POP more focus/emphasis is on data rather than procedures.
- OOP uses bottom-of-program approach for program design.
- Data and its operations in OOP are bound together into objects.
- Both OOP, objects can communicate with each other through message passing.
- In OOP data is hidden and cannot be accessed by external functions.

Scanned with CamScanner

- In OOP, data structures are designed such that they characterised objects.
- Object-oriented languages is very useful in application areas like real time, AI, expert systems etc.....

OOP vs POP

Object oriented programming	Procedure oriented programming
(1) In object oriented programming language is object oriented.	(1) Procedure oriented programming language is procedure oriented.
(2) In OOP, main focus is on data.	(2) In POP, main focus is on procedures.
(3) OOP follows, ~~bottom-up~~ top-down programming approach.	(3) POP follows, ~~top-down~~ bottom-up programming approach.
(4) In Basic Building block of OOP is object. OOP program is divided into number of meaning ful units known as objects	(4) In Basic Bulding block of POP is function. POP program is divided into number of functions or procedures.
(5) In OOP, object is collection of data and members.	(5) In POP, function is collection of instruction that perform specific task.
(6) In OOP, importance is given to data rather than procedures or function because it works as real world	(6) In POP, importance is not given to data but to the functions.
(7) OOP provides data hiding. So, it offers more security.	(7) POP does not have any proper way for hiding data so it is less secured.

(7) In OOP, objects communicate through member functions.	(8) In POP, functions communicate through arugements.
(9) OOP is an data centric language.	(9) POP is an Procedure centric language.
(10) examples. C++, Java, VB.Net.	(10) example, C, VB, DASCAL, FORTRAN.

* Basic concepts of OOP.

- object
- classes

} main focus

- Data encapsulation & abstraction.
- polymorphism / Inheritance
- dynamic binding.
- message passing.
- object orientation
- Application of OOP.

Inheritance
Modification
use inheritance

poly → many
morphism → forms

Question

(1) Applications of OOP - 4m / 2m
(2) What is class & object ? (Define) / 2m
(3) Describe following term - inheritance, data abstraction, data encapsulation, dynamic Binding. (4m)
(4) List any four object - oriented languages / 2m
(5) Enlist any four concept of OOP. / 4m.

Scanned with CamScanner

Page No.
Date

Q. List four object-oriented languages

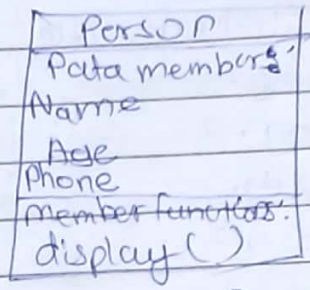
⇒ (1) C++
(2) Java.
(3) Python
(4) Ada
(5) C++
(6) Object Pascal.
(7) C#, B.net.
(8) Turbo Pascal.
(9) Simula.

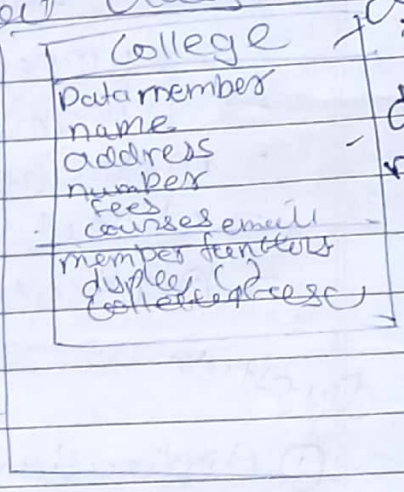
7/8/23

Q. Define: class & object

⇒ class

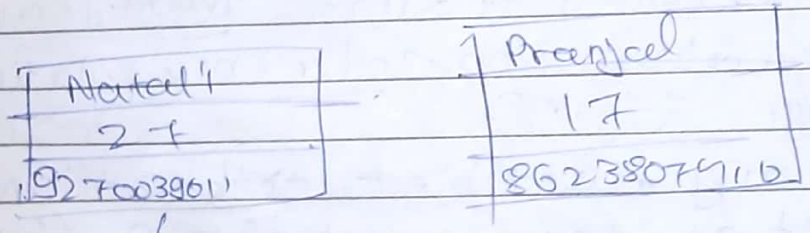
* A class is a way of grouping objects having similiar characteristics. characteristics
* A class is a user defined data type which represents a group of similiar objects.
* To create an object of a particular class we need to specify data members and member functions and member functions of that class
* Example:





class - Person.

* This shows the a class with data member, name, age, phone. with member function display()



objects of class person.

Page No.
Date

Object of College

SITM
Palghar
721xxxx
30k
compmechcivil
sjcem@gmail.com

SDSM
Palghar
802xxxx
40k
commerce,Arts
sdsm2020@yahoo.com

Object of class Person

Objects

1. Objects are basic runtime entities (groups) any real world object can be treated as an object in OOP. Such as person, car, etc.....
2. An object is an instance of class.
3. Object consist of two characteristics namely state (attribute) and behaviour (operation)
4. The state of an object is one of the possible conditions that an object can exist and it is represented by it's characteristics while behaviour of an object determines how an object behaves and it is represented by the operations it can performed.
5. Represent an object

ObjectName
State-1 State-2 . State N
Behaviour-1 . Behaviour N.

(Represented

Example (object of class Person

Object P1
State 1: Name State 2: Add Behaviour-1 - Display()

Scanned with CamScanner

Page No.
Date

Class - Employee

Employee
Data members: name: phoneno: department salary: add:
member functions: display() enter phone() enter dept()

Object

Prajwal Desai
8623××××
Dep
15k
Mumbai East

Reshmi Raut
9823××××
mech
20k
Andheri (W)

Class - student

Student
Name: Std: Div: Dept: Add:
display() enter std() enter Div() enter Add() enter Dept()

Object

Shruti Patil
SY
B
COMPS
Andheri East

Rahul Panchal
TY
C
mech
Virar (W)

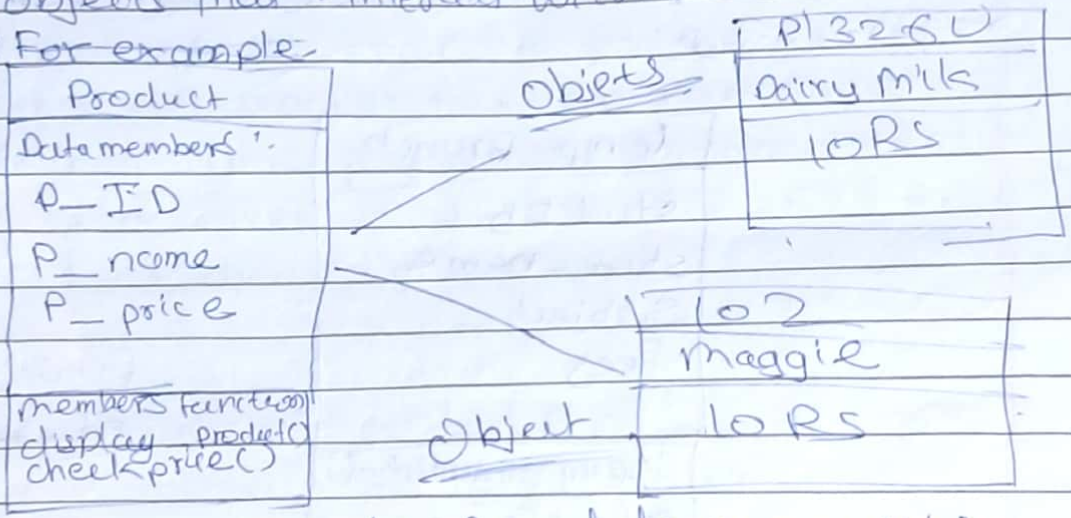
2/8/2023

Page No.
Date

Q. Describe the following terms :-
Enlist any 4 operations of OOP

① Data encapsulation and Data abstraction.

⇒ * Encapsulation is a technique of binding data and functions in a single unit called a class.
* Data encapsulation is a way to implement data abstraction in OOP.
* It provide services to external functions or other objects that interact with it.
* For example:



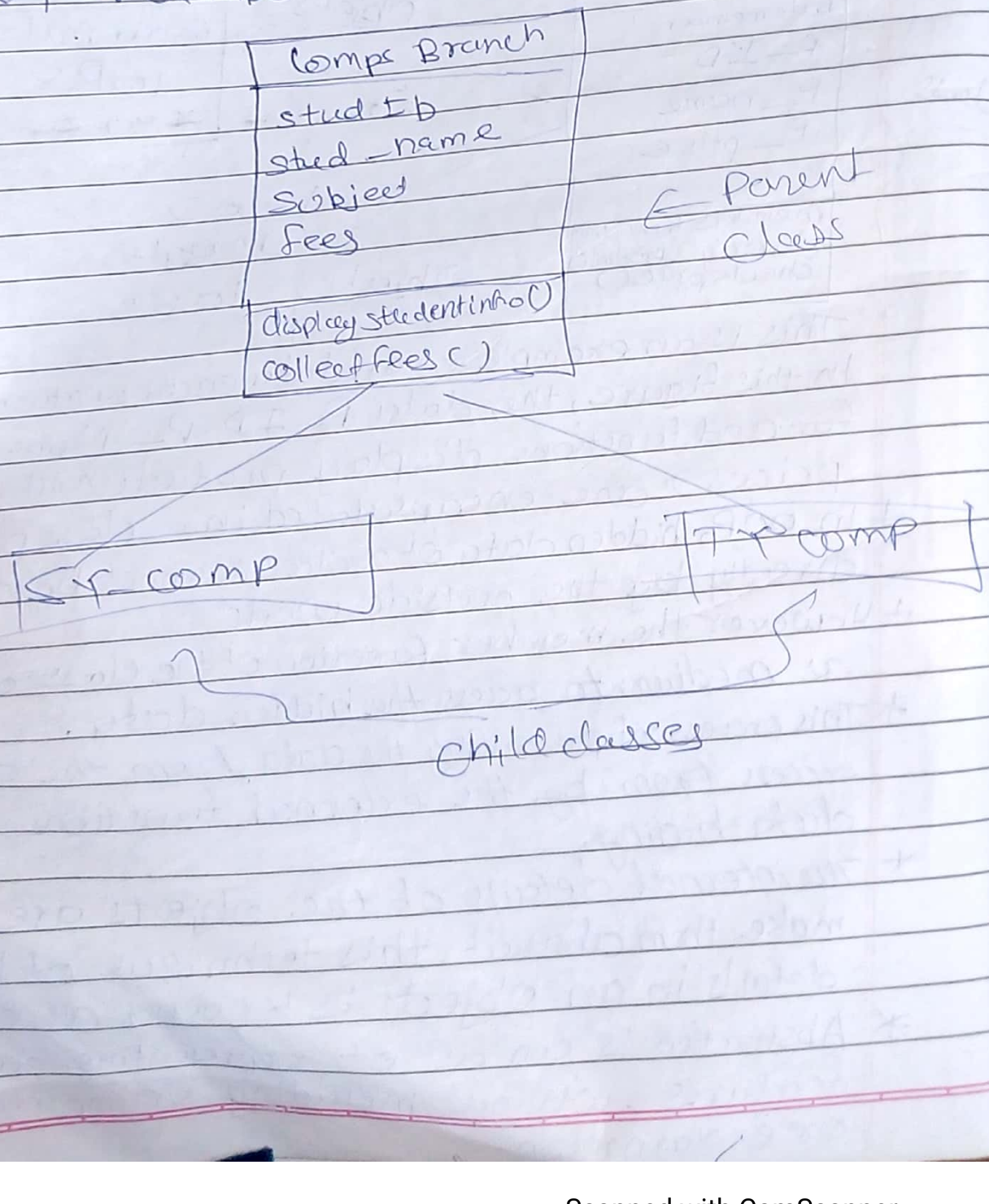
This is an example of an data encapsulation.
In this figure, the data P_ID, P_Name, P_price and functions display product() and check price() are encapsulated in a class product.
* In OOP, hidden data of a class cannot be accessed directly by the outside world.
* However the member function of the class act as medium to access the hidden data.
* This process of preventing the data from the direct access from by the external function is called data hiding.
* The internal details of the objects are hidden, which make them abstract. this technique of hiding internal details in an object is known as data abstraction.
* Abstraction is an act of representing ece essential features without including unimportant aspects or explanation.

(2) Inheritance

⇒ (*) The ability of object of one class to acquire properties of objects of another class is known as Inheritance.

(*) The existing class is known as base class (parent class) and the new class derived from exist class is known as derived class (child class)

(*) For example

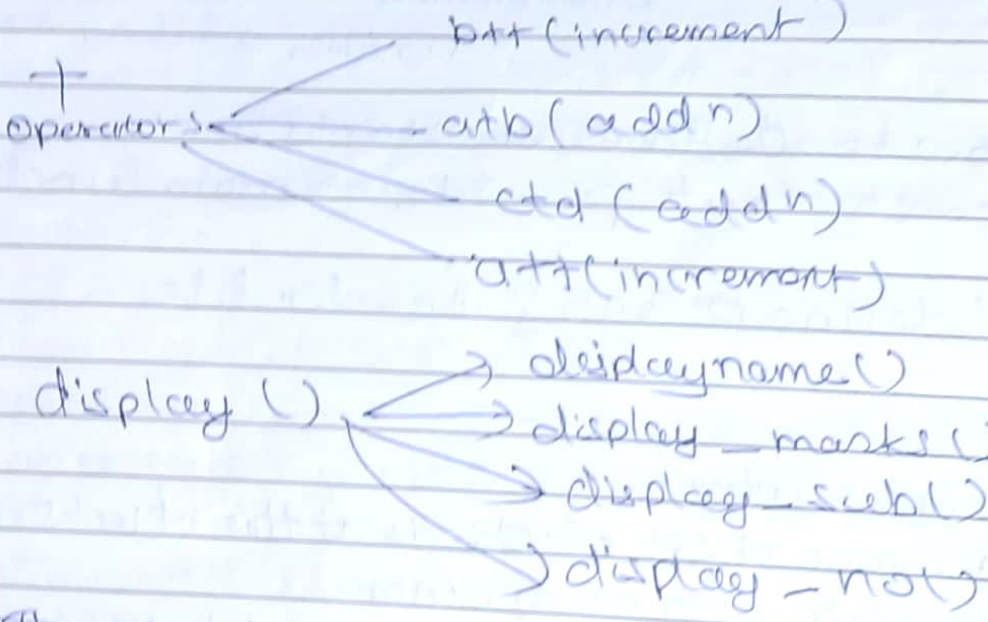


classes

(example (Hello world))

(3) Polymorphism

* In the term polymorphism poly stands for many and morphism means form. It means many forms having.
* Polymorphism is the ability of an entity. such as functions or a message to process in more than one form.
* Polymorphism in c++ is can be achieved either at compiler time or run time.
* At compile time polymorphism is implemented using operator overloading (operates using same name) and function overloading.



* The operators are. * From single operator many operations can be performed example shown above (+) (increment, addition etc)
* From single function (main) many other type functions can be derived example as shown above.

(4) Dynamic Binding

⇒ • Binding refers to linking of a procedure called to the code to be executed in response to the call.
• Dynamic Binding is also called late Binding
• Means that code associated with a given function call is not known until call at the run time.
• C++ language implements dynamic Binding through the use of virtual functions.
• Dynamic Binding is the process of linking function call to the actual code of the functions at run time.

function link → actual code
Dynamic Binding

(switch case)

5/8/23

Q How to define constants in C++.

⇒ 1) const float pi = 3.14 → main function.

2) #define PI 3.143 → header file

(5) Message passing:

⇒ ① OOP consist of set of objects of this objects communicate with each other.
② message passing is a process by which different objects in a program interact with each other that is when the program is executed objects send the object interact and communicate by sending and receiving messages.
③ The message in OOP are exchanged by calling the member functions of the classes.

Page No.
Date

(4) Working:
Step 1: Create classes for defining objects.
Step 2: Declare required objects.
Step 3: Establish communication among objects through message passing

(5) A message in OOP is interpreted as a request for execution of function.

(6) Example:

employee . salary (emp_id)
↑ ↑ ↑ ↑
object communicator message information/Data.
operator

Here an employee is nothing but object sending message to salary fun to find salary secured by employee with secified emp-id. In this case, function called salary is treated as message and parameter (emp_id) is information passed to object.

Q// Applications of OOP.

(1) Simulation & modelling - Simulation is an technique of representing real world entities with the help of computer program.
Simula 67 & smalltalk are two OOP languages which are designed for making simulation.

(2) Scripting
OOP has also been used for developing HTML, XHTML, XML Languages. Python, Ruby, Java are the languages based on object oriented Principles which are used for scripting.

(3) User Interface design:
This is popular application of OOP in the area of designing GUI such as windows.
C++ is used for developing user interfaces.

④ Object database.
- This databases store the data directly in the form of objects.

⑤ Developing computer games and virtual reality.
OOP is used for developing computer games such as
1) Diablo, 2) War Craft III
These game offers virtual Reality environment in which a number of object interact with each other.

⑥ Some other areas of application.
- AI
- Expert System.
- Real time
- Neural Networks
- Parallel Programming etc.

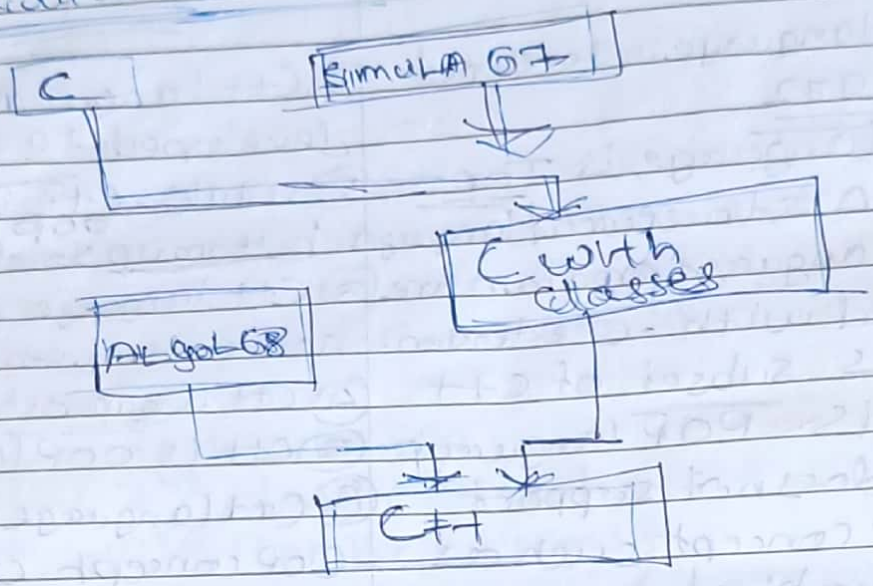
7/8/2023

Introduction to C++
- evolution of C++
- features of C++
- C vs C++
- Structure of C++

Questions:
① With suitable diagram describe Structure of C++ program. (4 marks)
② ~~Write down~~
⇒ ~~Evolution of C++~~

Page No.
Date

Evolution of C++



Features of C++

- C++ is simple
- It is portable
- Structured programming language (It structured programming language because we can break the program into parts using functions)
- Memory management. (C++ supports dynamic memory allocation)
- Speed - The compilation and execution time of C++ language is fast.
- pointer - using pointer in C++ we can directly interact with the memory we can use structures, functions, array etc.
- recursion - In C++ we can call the function within the function. It provides code reusability.
- Object oriented
- Compiler based. (C++ means without compilation no C++ program can be executed)

C	C++
(1) C language is developed in 1972	(1) C++ language is developed in 80.
(2) C language is Top-down structured language	(2) While C++ is OOP bottom up structured language.
(3) C language program are saved with .c extension.	(3) C++ language program are saved with .cpp extension.
(4) C is subset of C++	(4) C++ is superset of C
(5) C is POP language.	(5) C++ is OOP language.
(6) C doesn't support OOP concept such as class, object, encapsulation, inheritance, polymorphism, abstraction etc.	(6) C++ language supports OOP concept class, object, encapsulation, inheritance, polymorphism, abstraction etc.
(7) C language is middle level language.	(7) C++ language is higher level language
(8) In C language header file required is <stdio.h>	(8) In C++ language header file required is <iostream.h>
(9) In C language, scanf and printf is used as input and output fun^n.	In C++ language cin and cout is used as I/O fun^n

(10) #include <stdio.h>
void main
{
printf("Hello world");
}
	(10) #include <iostream.h>
	void main
	{
	cout<<"Hello world";
	}

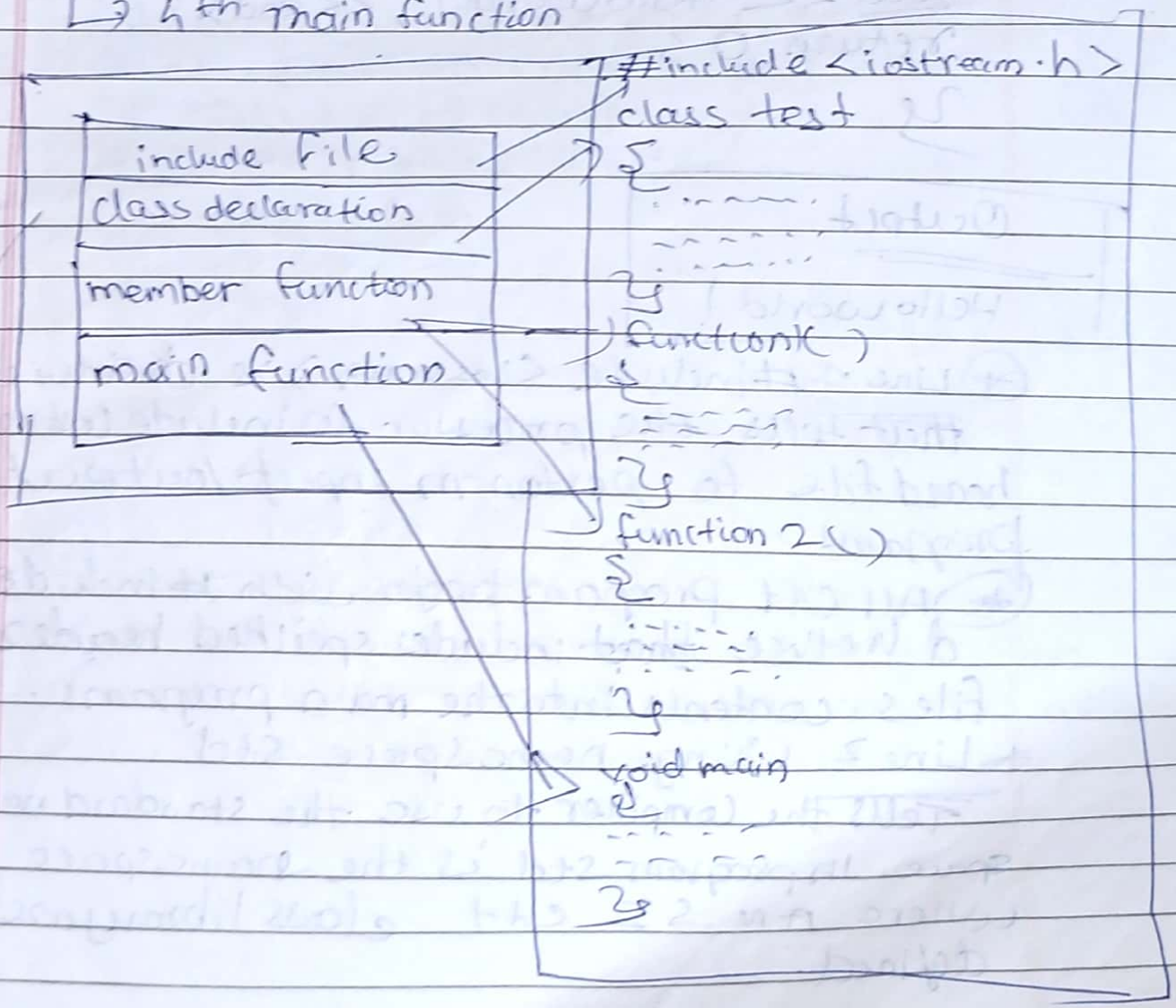
(11) Support does not Operator Overloading	(11) Support operator overloading

n/oct/2023

Page No.
Date

Structure of C++ program

① A C++ program is a text file containing a sequence of C++ commands (instructions)
② Put together according to C++ syntax rules.
③ C++ text file is known as the source file.
④ C++ source file carries file extension .cpp.
⑤ A typical C++ program would contain following 4 section -
- 1st Include file.
- 2nd class declaration.
- 3rd Member functions
- 4th main function



Page No.
Date

Above figure shows structure of c++ program. c++ program is collection of function and list of library files

Consider the following hello world program which shows basic structure of c++ program.

```
#include <iostream.h>
using namespace std;
int main()
{
cout << "Hello world!" << endl; //
return 0; //
}
```

Output

```
Hello world!
```

Line 1 #include <iostream> is directive that tells the processor to include iostream header file to perform input/output programs.

All c++ program begin with #include directive that includes specified header files contents into the main programs.

Line 2 Using namespace std

Tells the compiler to use the standard name space. In program std is the namespace where ANSI c++ class library are defined.

* cout and endl which are used in program belong to ~~std~~ namespace std;
* Line 3 int main() declares main function. which is the point from where all c++ program begin their execution.
* Every program should have main function.
* Line 4: Contains opening curly braces "{" which marks the begining of body of program.
* Line 7: Contain closing curly braces "}" which marks end of body of program.
* Line 5: cout << "Hello World" << endl.
  is output statement.
  cout refers to console output.
  << ⇒ Is called insertion ~~operation~~ operator. to put the string "Hello World" in console.
  >> ⇒ Extraction operator
  they are used to read data from input device.
  endl ⇒ newline or end of line.
* Line 6: return 0;
  It's called as return statement.
  This return statement returns the value 0 to Operating system which means everything went "Okay" means programming successful.

many error

Page No.
Date

Basics of C++ - 1.4

Question

Q// Explain different operators used in C++. (4m)

Q// Describe concept of type casting using suitable example. (4m)

Q//

1.5 Control structures

Control structure
- if
- if-else
- nested if
- switch
- loops (for, while, do-while)
- loop control statements
  - break
  - continue
  - goto
- Programs related to ⇒ this

Questions

④ Write C++ program to find factorial of given number using loop. (4m)

09/08/2023

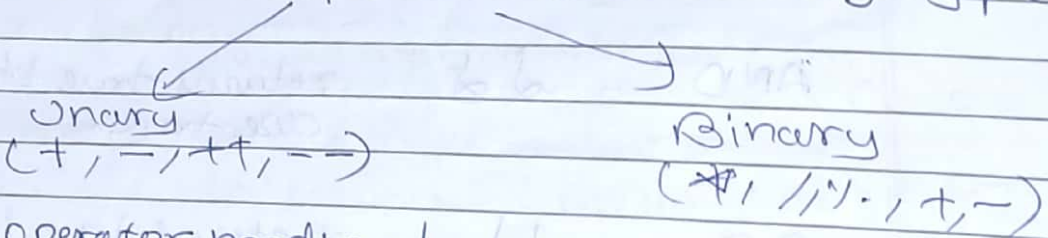
Page No.
Date

Operators in c++

→ Operators instructs c++ to perform some operation on one or more operands. a+b operands operator operands.
For example a+b
'a & b' are operands and '+' is operator.

→ in c++, operators can be classified into following category

1. Arithematic Operators.
   Arithematic operators used to do arithematic operation such as (addition, subtraction, division, multiplication etc)
   Arithematic Operators have following type.



* Unary operator needs only one operands.
* Binary operator needs two operands.

Example
Addition 'a+b' '+' operator. } 'a, b' operands.
Subtraction 'a-b' '-' operator }
Mul 'a*b' '*' operator
Div 'a/b' '/' operator.

* increment 'a++' (increase value by 1)
  decrement 'a--' (decrease value by 1)

2. Relational Operator
This operator are used to compared 2 values.
<= (less than operator)
<= (less than equal to)
>= (greater than)
>= (greater than equal to.
== (equals to)
!= (not equal to)

many compiler
Overloaded

Page No.
Date

example
a < b (a is less than b)
a > b (a is greater than b)
a <= b (a is less than equals to b)
a >= b (a is greater than equals to b)
a == b (a is equals to b)
a != b (a is not equal to b)

③ Logical operator
logical operator in c++ combines the result of two or more than two expressions.
Logical expression returns 1 if the result turns and it returns 0 when result is false.

"AND" "&&" returns true if the conditions are true.

OR || returns true if atleast one condition is true

Not ! → !True → False
!False → True

④ Assignment Operator
Assignment operator are used to assigned left value to operands/variable - left value to right.

Operators	Description
=	Symbol equals to
+=	plus e
-=	
*=	
/=	

Operators	Descriptn	Ex:
=	Simple assignment Operator assign value from right side operand to left side operands.	c = a + b c will assigned value of a + b
+=	add and assignment operator this adds right operand to left operand assign result to left operand	(c += a) c = c + a it acts right operands to the left operand c [c = a + c]
-=	Sub and assignment operator - this sub right operand to left operand assign result to left operand	(c -= a) [c = a - c]
*=	Multiply AND assignment Operator - this multiply right operand to left operand assign result to left operand	c *= a [c = c * c]
/=	Divide AND assignment operator this divide right operand to left operand assign result to left operand.	c /= a c = c / c

10/08/2023

Any datatype, logic Bitwise - Bits

Page No.
Date

(5) Bitwise Operator

* Bitwise operators in C++ refers to the sting setting or shifting Actual bits.
* They are used to manipulate bits name of operator Symbol Description.

Example			Example
~~Example~~			
(1) AND	&	Logical AND Perform	0101 & 0011 = 0001
(2) OR	|	Logical or	0101 | 0011 = 0111
(3) Shift Left	<<	To shift bits to left	00010111 << 1 = 00101110
(4) Shift right	>>	To shift bits to right	00010111 >> 1 = 00001011
(5) Complement/ Not	~	to negate each bit	~0001 = 1110

Typecasting

i) Type casting means conversion of one data type to another data type.
ii) Example, float value can be converted into integer value.
iii) There are two types of "type conversion" in C++
- automatic conversion / implicit conversion
- type casting / explicit conversion.

Implicit Type

(i) In this the conversion is done automatically by the compiler.
(ii) In this smaller data type is converted into larger data type
(iii) Hierarchy is int → unsigned int → long → unsigned long → long → ~~float~~ double → long double.
(iv) In most operations, operands require to be of same type if they are not then compiler will convert them using above hierarchy.
(v) When compiler tries to convert lower level data type to higher level data type then it

it is known as widening or Promotion.

(vi) When compiler tries to convert Higher level data type to lower level data type then it is known as Narrowing or Coercion.

(vii) Example :- double d = 2.3;
int i;
i = d;
printf("%d", i);

O/P = 2

Explict Type

(i) it refers to the type conversion that is performed explicitly

(ii) this can be performed by using two different forms.

- datatype (expression) → int (2.3)
- (data type) expression → (int) 2.3;

data type is also known as cast operator
Exp - (int) 20.23

(iii) In above example, the value 20.23 is converted into integer value 20.

(iv) Here (int) is used as type operator to convert float value to integer value.

operands	&	|	(or)	~
T T	T	T		
1 1	1	1		T → F
F T	F	T		
0 1	0	1		F → T
T F	F	T		1 → 0
1 0	0	1		0 → 1
F F	F	F		
0 0	0	0		

many can be overloaded

* WAP in c++, to check whether the number is Even or odd

```
#include <iostream.h>
#include <conio.h>
void main()
{
int n;
cout << "Enter number";
cin >> n;
if (n%2==0)
  {
  cout << "No is even";
  }
else
  {
  cout << "No is odd";
  }
getch();
}
```

21/08/2022

Page No.
Date

WAP to fing largest of 3 number

```
#include <iostream>
using namespace std;
int main()
{
 int a, b, c, big;
 cout<< "Enter three numbers" ;
  //assuming 'a' as largest number.
 big = a;
 if (b>big)
  {
   big=b;
  }
 if (c>big)
  {
   big=c;
  }
cout << "Largest number is = " << big;
return 0;
}
```

Output

```
Enter three numbers
20 30 69
Largest number is 69
```

* WAP to display multiplication table of 7.
(5 marks)

```
#include<iostream>
using namespace std;
int main()
{
int n,i;
cout<<"Enter a number";
cin>>n;
for(i=1;i<=10;i++)
{
cout<<n<<"*"<<n*i<<endl;
cout<<"n"<<"x"<<"
cout<<n<<"x"<<i<<"="<<n*i<<endl;
}
return 0;
}
```

Output

Enter a number: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Page No.
Date

WAP to factorial of number

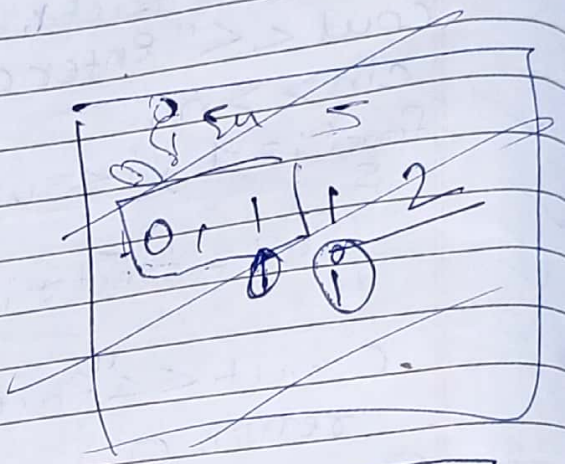
```
#include <iostream>
using namespace std;
int main()
{
int n, i, fact=1, feet=1;
cout<<"enter a number";
cin>>n;
for(i=1;i<=n;i++)
  {
   feet=feet*i;
  }
 cout<<"factorial of number is:"<<fact;
  return 0;
}
```

Output

```
Enter a number 3
Factor of number is 6
```

* WAP to print fibonacci series upto n term's

```
#include<iostream>
Using namespace std;
void main()
{
int i, f1, f2, f3;
f1=0
f2=1;
cout<<f1;
cout<<f2;
cout<<"enter n";
cin>>n;
for(i=1; i<=n; i++)
{
f3=f1+f2;
cout<<f3;
f1=f2;
f2=f3;
}
}
```



Output

0, 1, 1, 2, 3, 5

17/05/23

Page No.
Date

Scope Resolution Operator (::)

1. C++ language supports a mechanism to access global variable from a function in which a local variable is defined with the same name as a global variable.
2. It is achieved using scope resolution operator.
3. The scope resolution operator is denoted by pair of (::)
4. The syntax is
   :: variable_name;
5. Program for Scope Resolution Operation

```
#include <iostream>
Using namespace std;
int a=10; //global variable.
int void main()
{
int a=15; //Local variable
cout<< "Local a=" << a << "Global a = " << ::a;
return 0;
}
```

Output

Local a=15 Global a=10.

global

pointer - memory location

Page No.
Date

Memory management operator :-

* In C++ language for dynamic allocation of memory, we can use new and delete operators performs.
* This two operators perform task of allocating and free the memory in easier way.
* The keywords, new and delete can be used to allocate and deallocate memory this.
* This operators are memory management operator's also known as free store operator.

The new operator.
→ The new operator is unary operator that allocates memory and returns a pointer at the starting address of allocated space.
Syntax : ptr_var = new data_type
Example :
```
int *p;
p= new int(55);
```

The delete operator
→ The delete operator deallocates, frees the memory allocated by new.
Syntax
```
delete ptr_var;
```
Example
```
delete p;
```

Page No.
Date

Array, string, structures in C++.

Questions

Q1 C++ program to find and display smallest number of an array (4marks)

Q2 ~~Write a program to accept array of five elements, find and display smallest number from it.~~ (4marks) OR

Q2 WAP to solve 1-D array in ascending order. (4marks)

Q3 WAP to add two 3 x 3 matrics and duplay addition. (4marks)

Q4 WAP to declare a struct book with members book id, book name, accept and display data of 5 books using array of structure.

Q5

many error
Overloaded not

Page No.
Date

* Write C++ program to print array of 5 numbers

```
#include <iostream>
using namespace std;
void main()
{
int a[5], i;
cout<<"Enter 5 numbers";
for(i=0;i<5;i++)
{
cin>>a[i];
}
cout<<"Array of elements";
for(i=0;i<5;i++)
{
cout<<a[i];
}

}
```

Output

Enter 5 numbers.
10 15 20 25 30
Array of element.
10 15 20 25 30.

```
#include <iostream>
using namespace std;
void main()
{
int a[30], n, i, small;
cout << "Enter number of element in array\n";
cin >> n;
cout << "Enter element in array";
for(i=0; i<n; i++)
 {
 cin >> a[i];
 }
 small = a[0];
 for(i=0; i<n; i++)
 {
  if(a[i]<small)
    {
    small = a[i]
    }
  }
cout << "Smallest element = " << small;
}
```

21/08/23

Q) WAP to write Ascending order using Array.

```
#include<iostream>
Using namespace stand std;
void main()
{
int a[30], n, i, j, temp;
cout << "Enter the total no of element :";
cin >>n ;
cout<<"Enter elemants of arrey :" \n";
for(i=0; i<n ; i++)
{
  cin >> a[i];
}
for(i=0;i<n;i++)
{
  for(j=1;j<n;j++)
  {
    if(a[i] > a[j])
    {
    temp= a[i];
    a[i] = a[j];
    a[j] = temp;
    }
  }
}
cout<<"Array in assending order";
for(i=0;i<n;i++)
{
  cout<<a[i];
}
return 0;
}
```

Page No.
Date

Output

Enter total no of el 5

Enter element 10 20 3 5 6

Array in ascending order

3 5 6 10 20

WAP

3×3 matrices addition.

```
#include <iostream>
void main()
{
int a[3][3];
int b[3][3];
int c[3][3];
int i, j;
cout << "Enter elements of A";
for(i=0; i<3; i++)
{
  for(j=0; i<3; j++)
  {
    cin >> a[i][j];
  }
}
cout << "Enter elements of B";
for(i=0; i<3; i++)
{
  for(j=0; j<3; j++)
  {
  cin >> b[i][j]
  }
}
}
```

a_{01} a_{11} a_{21}
a_{02} a_{12} a_{22}
a_{03} a_{13} a_{23}

```
for(i=0; i<3; i++)
  {
  for(j=0; j<3; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
cout << "Addition =\n";
for(i=0; i<3; i++)
  {
  for(j=0; j<3; j++)
  {
    cout >> c[i][j]
  }
  -> cout<< "\n";
}
return 0;
}
```

output

c ↓ ⇒

10	20	30
30	40	50
60	70	80

1	1	1
1	1	1

11	21	31
31	41	51
61	71	81

28/08/23

WAP to declare structure of book with members, book_id, book_name, accept and display data of 5 books using array of structure.

```
#include<stdio.h>
#include<iostream>
using namespace std;
struct book
{
int book_id;
char name[50];
}b[10];

void main()
{
cout<<"Enter info";
for(int i=0;i<5;i++)
{
cout<<"enter id";
cin>>b[i].book_id;

cout<<"enter name";
cin>>b[i].name;
}
cout<<"Display info";
for(int i=0;i<5;i++)
{
cout<<"\n book id"<<b[i].book_id;
cout<<"name="<<b[i].name;
}
}
```

Output

Enter info
enter id: 1
enter name: abc

enter id: 2
enter name: xyz

enter id: 3
enter name: bbc

enter id: 4
enter name: ssx

enter id: 5
enter name: TTT

Display info
Book id is 1
name: abc
Book id is 2
name: xyz
Book id is 3
name bbc
Book id is 4
Name ssx
enter id 5
name TTT.

Q// Accept data for 5 student and display it using structure.

```
#include<iostream>
using namespace std;
struct Student
{
int roll_no;
char name[50];
}s[10];

int main()
{
cout<<"Enter info:"<<endl;
for(int i=0;i<5;i++)
{
cout<<"enter id: Roll no"<<endl;
cin>>s[i].roll_no;

cout<<"Enter name:"<<endl;
cin>>s[i].name;
}

cout<<"Student info";
for(int i=0;i<5;i++)
{
cout<<"In books id is"<<s[i].roll_no;
cout<<"name:"<<s[i].name
}
return 0;
}
```

3/08/2023

Page No.
Date

// Find the sum of an array's elements.

```
#include<iostream>
using namespace std;
void main()
{
int arr[10], i, sum=0;
cout<<"Enter 10 array elements";
for(i=0;i<10;i++)
  {
    cin>>arr[i];
  }
for(i=0;i<10;i++)
{
sum=sum+arr[i];
}
cout<<"Sum of all array elements = "<<sum;
cout<<endl;
return 0;
}
```

Output

Enter 10 array elements

1 2 3 4 5 6 7 8 9 10

Sum of all array elements = 55