Expt. No..... Date .....

1. Local variables are stored in an area called Stack.

2. B

3. When the inheritance is private methods in base class are Inaccessible in the derived class (in C++).

4. D. Programmer have to always call destructor at the end of the program.

5. True.

(2) a New operator denotes a request for memory allocation on free store.

Syntax

→ pointer-variable = new data-type;

Ex. int *P = NULL;
P = new int;

Delete operator. used to deallocate dynamically allocated memory.

Syntax: → delete pointer-variables;

Ex. delete P;

Expt. NO..... Date .....

2.b. A Constructor is a Special type of function with no return type.
without a constructor, you can't create instance of the class.
there are 2 type's. → Construct u{ a=10; b=20; }
· Default constructor. Point (int x1, int y1){
· Parameterized constructor → x=x1;
· Copy Constructor → y=y1; }
ClassName (const ClassName &old_obj);

2. C.

Procedural Oriented programming	object oriented Programming
1. In procedural programming, Program is divided into small parts called function	In object oriented programming Program is divied into small parts called objects.
2. Procedural programming follows top down approch.	object oriented programming follows bottom up approch.
3. There is no access Specifier in procedural programming.	it having access specifiers like private, public, protected etc.
4. Adding new data and function is not easy	Adding new data and function is easy.
5. Ex. C, Fortran, Pascal, Basic etc	C++, Java, Python, C#. etc.

Teacher's Signature

public: void disp(){ obj2. disp();
return 0;

Expt. No..... Date .....

Long!

(11)

```
#include <iostream>
using namespace std;
void Sort012 (int a[], int arr_size)
{
    int lo = 0;
    int hi = arr_size - 1;
    int mid = 0;
while (mid <= hi) {
    switch (a[mid]) {
case 0:
        swap (a[lo++], a[mid++]);
        break;
Case 1:  mid++;
        break;
Case 2:  swap (a[mid], a[hi--]);
        break;
    }}}
void printArray (int arr[], int arrsize, Size)
{ for (int i=0; i<arr_size; i++) cout << arr[i] << " ";
} int main() { int arr[] = {1122002112};
int n = sizeof (arr) / sizeof (arr[0]);
Sort012 (arr, n);
    cout << array after Segregation";
    PrintArray (arr, n);
        return 0;
    }
```

output 00111222

Expt. No..... Date .....

```
#include<iostream>
#include<string>
using namespace std;
class member{
 char name[20], address[40];
 double number; int age;
public:
   int &Salary;
    void input(){
    cout<<endl;
 cout<<"Name:"<<endl;
 cin.getline(name,20);
 cout<<"Age".<<endl;
  cin>>age;
 cout<<"phone Number:"<<endl;
  cin>>number;
  cout<<"Address:"<<endl;
  cin.getline(address,40);
  cout<<"Salary:"<<endl;
  cin>>Salary;}

void display(){
          cout<<endl;
cout<<"name:"<<name<<endl;
   cout<<"age"<<age<<endl;
   cout<<"address"<<address<<endl;
   cout<<"Salary:"<<Salary<<endl;}};
```

Teacher's Signature

Expt. No..... Date .....

```
class employee : public member {
 char specialization [20], department [20];
public:
 void input () {
cout << "\n\t Enter Employee Details \t \n";
member :: input ();
cout << " Specialization : " << endl;
cin.getline ( specialization 20);
 cout << " Department : 'endl;
cin.getline (department ,20 ); }
 void display () {
  cout << "\n\t Displaying Employee Details \t \n";
  member :: display ();
  cout << "Specialization : " << specialization << endl;
cout << "Department : " << department << endl; }
void paint Salary () {
  cout << "\n Salary of the member is : "
  Salary << endl; } };

class manager : public member {
   char specialization [20], department [20];
  public:
   void input () {
cout << "\n\t Enter manager Detail \t \n";
   member :: input ();
   cout << " Specialization : " << endl;
   cin << cin.getline ( specialization, 20);
```

Expt. No.....

Date .....

```
cout << "Department : "<<endl;
 cin.getline (department,20); }
void display ()
{
cout <<"\n\t Displaying manager Details \t\n";
member :: display();
cout <<"Specialization : "<< specialization <<endl;
cout << "Department : "<< department <<endl;
}
void printsalary () {
cout <<" \n Salary of the member is : "salary<<endl;
}};
int main () {
employee e;
manager m;
e. input ();
m. input ();
e. display ();
e. print salary ();
m. display ();
m. printsalary ();
}
```