1) ~~Stack~~

Name:- S.V. Sumanth kumar Reddy

1) stack

2) Compile error in line "Derived *dp = new base;"

3) Inaccesible

4)

5)

Short Answer Questions:-

1) C++ Supports dynamic allocation and dellocation using the new and delete operators The operators allocate memory for objects from a pool called the free store. The new operator calls the special function operator new and the delete operator calls the special function operator delete.

```
#include<iostream>
using namespace std;
class student{
 int age;
 public:
student(): age(12) { }
 void get age() {
 cout<<"Age:"<<age<<endl; }
};
int main() {
student *ptr = new student();
ptr -> get age();
```

```
delete ptr;
return 0;
}
```

Output :-

Age = 12

2) The Constructor which creates an object by copying variables from another object is called as copy constructor. The purpose of copy constructor to Initialize a new instance to the value of existing instance.

Different type of constructors :-

→ Default Constructor
→ Parameterized constructor.
→ Copy constructor.

3) Diff b/w Procedural oriented Programming and object oriented Programming.

Procedural oriented Programming	object oriented Program
1) Procedural Programming follows top down approach	1) Obj. oriented Program follows bottom up approch
2) This is no access specifier in Procedural Programing	2) Obj. oriented Programing have access specifier like Private, Public, Protected etc...
3) Adding new data and function is easy	3) Adding new data and func is easy.

4) Procedural program based on unreal world

Ex:- C, Pascal, Basic, etc

4) Obj. oriented Program based on real world.

Ex:- C++, Java, Python.

Long Answer Questions:-

1) Explain the types of Polymorphism. Polymorphism is constructed as one of the important features of Obj oriented Programming in C++ Polymorphism is mainly divided into two types

i) Compile time Polymorphism

ii) Run time polymorphism.

Code for polymorphism:-

```
Class Animal {
Public:
void animalsound() {
cout << "The animal makes a sound.\n";
}
};
Class Pig : Public Animal {
Public:
void animalsound() {
cout << "The Pig says: wee wee \n"; }
};
Class Dog : Public Animal {
Public:
void Animal sound() {
cout << "The dog says: Bow wow \n";
```

```
}
};
```

B)
```
#Include<stdio.h>
Void swap (int A[], int i, int j) {.
int temp= A[i];
 A[i] = A[j];
 A[j] = temp; }
int threeway Partition(int A[], int end) {
int start=0 , mid= 0;
 int pivot = 1;
while(mid<=end) {
if (A[mid] <pivot)
{
swap(A, start, mid );
 ++start, ++mid;
}
else if (A[mid] >pivot)
{
swap (A, mid, end);
 ---end;
}
else {.
++mid;
}
}
};
```

```
int main ( )
{
int A[ ] = {1, 1, 2, 2, 0, 0, 2, 1, 2};
int n = sizeof(A) / sizeof(A[0]);
threeWayPartition(A, n-1);
for (int i = 0; i < n; i++) {
 Printf ("%d", A[i]);
 }
return 0;
}
```

```
① #include <iostream>
#include <string>
Using namespace std;
char name[20], address[40];
 double number;
int age;
Public:
int Salery;
void input ( )
{ cout << endl;
cout << "Name:" << endl;
cin.getline(name, 20);
Cout << "Age:" << endl;
  cin >> Age;
cout << "Phone number:" << endl;
  Cin >> Agecnumber;
```

```
Cout<<"Adress: "<<endl;
cin.getline(address,40);
Cout<<"Salery: "<<endl;
 cin>>Salery
}
void display()
{ Cout<<endl;
Cout<<"Name :"<<name<<endl;
Cout<<"Age:"<<age<<endl;
Cout<<"Phonenumber :"<<number<<endl;
Cout<<"Adress," <<Adress<<endl;
Cout <<"Salery:"<<Salery<< endl;
}
};
Class employe: Public member {
Char specialization[20], department[20];
 Public:
void input()
{
cout <<"\n !! Enter employe details !! \n";
member::display();
Cout<<"Specialization :"<<specialization<<endl;
Cout <<"Department:"<<department<<endl;
}
void Print Salery() {
Cout<<"\n Salery of member is :"<<Salery<<endl;
```

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

```
void Print salery()
{
cout<<"\n salery of the number is :"<<salery<<endl;
}
};
int main()
{
employe e;
manage m;
e.input();
m.input();
e.display();
e.Print salery();
m.display();
m.print salery();
}
```