PART A

6.) Write a C++ program to create class called list with member functions to insert an element from front as well as to delete element from front of list. Demonstrate all functions by creating list object.

```
#include<iostream>
```

```
#include<new>
using namespace std;
struct nod
{
int info;
struct nod*next;
};
typedef struct nod node;
class list
{
node *f;
public:
list()
{
f=NULL;
}
void ins(int num)
{
node *p=new node;
p->info=num;
p->next=f;
f=p;
}
void del()
{
node *temp=f;
if(f==NULL)
cout<<"\nNo elements to delete.\n";
else
{
cout<<"\n The deleted elements is :\n"<<f->info;
f=f->next;
delete temp;
cout<<"\n Deletion successfull \n";
}
return;
}
void disp()
{
node *temp=f;
if(f==NULL)
```

```
cout<<"\n List is empty \n";
else
{
cout<<"\n Elements in the list are: ";
while(temp!=NULL)
{
cout<<" "<<temp->info;
temp=temp->next;
}
}
}
};
```

```
int main()
{
int num,ch=1;
list ob;
cout<<"\n!!!!!!!!!! LINEAR LINK LIST !!!!!!!!!!!!!!!\n";
cout<<"\n1] Insert\n2] Delete\n3] Exit";
while(ch)
{
cout<<"\nEnter your choice \n";
cin>>ch;
switch(ch)
{
case 1: cout<<"\n Enter no. to be insrted\n";
cin>>num;
ob.ins(num);
ob.disp();
break;
case 2: ob.del();
ob.disp();
break;
case 3: return 0;
default:cout<<"Invalid choice \n";
break;
}
}
}
```

7.) Write a C++ program to create a class called student with the dada members. USN, Name and Age using inheritance. Create classes UG student and PG student having field as semester, fees and stipend. Enter the data for at least 5 students and find semester wise average, age for UG and PG students respectively.

```
#include<iostream>
using namespace std;
class student
{
int reg,age;
char name[50];
public:
void getsdata()
```

```
{
cout<<"\nEnter name of student: ";
cin>>name;
cout<<"\nEnter regestration no.: ";
cin>>reg;
cout<<"\nEnter age: ";
```

```
cin>>age;
}
int giveage()
{
return age;
}
};
class ugstudent:public student
{
int sem;
float fee,sti;
public:
void getugdata()
{
getsdata();
cout<<"\nEnter Sem: ";
cin>>sem;
cout<<"\nEnter fee: ";
cin>>fee;
cout<<"\nEnter stipend: ";
cin>>sti;
}
int givsem()
{
return sem;
}
};
class pgstudent:public student
{
int sem;
float fee,sti;
public:
void getpgdata()
{
getsdata();
cout<<"\nEnter Sem: ";
cin>>sem;
cout<<"\nEnter fee: ";
cin>>fee;
cout<<"\nEnter stipend: ";
cin>>sti;
```

```
}
int givsem()
{
return sem;
```

```
}
};
int main()
{
int m,n,i,s;
cout<<"\nEnter the no. of UG student: ";
cin>>m;
ugstudent ug[m];
for(i=0;i<n;i++)
ug[i].getugdata();
for(s=1;s<=8;s++)
{
int flag=0,count=0,sum=0;
for(i=0;i<n;i++)
{
if(ug[i].givesem()==s)
{
sum=sum+ug[i].giveage();
flag=1;
count++;
}
}
if(flag==1)
{
cout<<s<<" sem avg. age is: "<<sum/cou;
}
}
cout<<"\nEnter the no. of PG student: ";
cin>>n;
ugstudent pg[n];
for(i=0;i<n;i++)
pg[i].getpgdata();
for(s=1;s<=8;s++)
{
int flag=0,count=0,sum=0;
for(i=0;i<n;i++)
{
if(pg[i].givesem()==s)
{
sum=sum+pg[i].giveage();
flag=1;
```

```
count++;
}
}
if(flag==1)
{
```

```
cout<<s<<" sem avg. age is: "<<sum/cou;
}
}
return 0;
}
```

---

8) (a) Write a C++ program for exception handling. Create a user defined exception classes for divide by zero and negative number input separately.

```
#include<iostream>
using namespace std;
class divide
{
int a,b;
public:
void
compute()
{
cout<<"\n
Enter two
integers
\n";
cin>>a>>
b;
try
{
if(b!=0)
cout<<a/b
<<"\n";
else
throw b;
}
catch(...)
{
cout<<"\n
Division
by
zero(error
) \n";
}
}
};
```

```
class neg
{
int a;
public:
void scan()
{
cout<<"\nEnter age \n";
cin>>a;
try
{
if(a>=0)
```

```
cout<<"\nAge of the person is "<<a<<" \n";
else
throw a;
}
catch(...)
{
cout<<"\nPlease enter positive value for age \n";
}
}
};
int main()
{
int c;
divide d;
neg n;
while(1)
{
cout<<"1.divide integers 2.enter age 3.exit \n";
cin>>c;
switch(c)
{
case 1:d.compute();
break;
case 2:n.scan();
break;
default:return 0;
}
}
return 0;
}
```

8(b) Write a C++ program for sorting names using file handling.

```
#include<iostream>
#include<fstream>
#include<cstring>
#include<cstdlib>
using namespace std;
int main()
{
char temp[20],name[20][20];
int i,j,n;
FILE *f;
```

```
f=fopen("sort.txt","w");
cout<<"\nEnter no. of names\n";
```

```
cin>>n;
cout<<"\nEnter "<<n<<" names\n";
for(i=0;i<n;i++)
{
cin>>name[i];
fprintf(f,"%s",name[i]);
}
fclose(f);
f=fopen("sort.txt","r");
if(f==NULL)
{
cout<<"\nFile doesnt exists\n";
return 0;
}
while(!feof(f))
{
fscanf(f,"%s",name[i]);
i++;
}
n=i-1;
cout<<"\nNames before sorting :\n";
for(i=0;i<n;i++)
cout<<name[i]<<" "<<endl;
cout<<"\nNames after sorting :\n";
for(i=0;i<n-1;i++)
{
for(j=0;j<n-i-1;j++)
{
if(strcmp(name[j],name[j+1])>0)
{
strcpy(temp,name[j]);
strcpy(name[j],name[j+1]);
strcpy(name[j+1],temp);
}
}
}
for(i=0;i<n;i++)
cout<<name[i]<<" "<<endl;
return 0;
}
```