

1. Write a program to illustrate the dynamic memory allocation.

```
#include<iostream>
using namespace std;
int main(){
    int i,size;
    int *ptr;
    cout<<"Enter how many number enter you want"<<endl;
    cin>>size;
    ptr=new int[size];
    for(i=0;i<size;i++){
        cout<<"Enter number"<<i+1<<endl;
        cin>>ptr[i];
    }
    cout<<"Entered number"<<endl;
    for(i=0;i<size;i++){
        cout<<ptr[i]<<endl;
    }
    delete[]ptr;
    return 0;
}
```

2. WAP to find the sum of n numbers entered by user using Dynamic Memory Allocation in C++.

```
#include<iostream>
using namespace std;
int main(){
    int i,size,sum=0;
    int *ptr;
    cout<<"Enter how many number enter you want"<<endl;
    cin>>size;
    ptr=new int[size];
    for(i=0;i<size;i++){
        cout<<"Enter number"<<i+1<<endl;
        cin>>ptr[i];
    }
    for(i=0;i<size;i++){
        sum=sum+ptr[i];
    }
    cout<<"The sum of n entered number is "<<sum;
    delete[]ptr;
    return 0;
}
```

3. Write a simple program using of dynamic memory allocation which should include calculation of marks of 3 subjects of n students and displaying the result as pass or fail & name, roll. Pass mark is 45 out of 100 in each subject.

```
#include<iostream>
#include<iomanip>
using namespace std;
class result{
    private:
        char name[30];
        int roll;
        float mark1,mark2,mark3;
    public:
        void getmarks(){
            cout<<"Enter the name of student"<<endl;
            cin>>name;
            cout<<"Enter the roll of student"<<endl;
            cin>>roll;
            cout<<"Enter the marks of 3 subject"<<endl;
            cin>>mark1>>mark2>>mark3;
        }
        void display(){
            if(mark1>45&&mark2>45&&mark3>45){
                cout<<"Name:"<<name<<endl;
                cout<<"Roll:"<<roll<<endl;
                cout<<"Result:PASS"<<endl;
            }
            else{
                cout<<"Name:"<<name<<endl;
                cout<<"Roll:"<<roll<<endl;
                cout<<"Result:FAIL"<<endl;
            }
        }
};

int main(){
    int n,i;
    result *ptr;
    cout<<"Enter the number of student"<<endl;
    cin>>n;
    ptr=new result[n];
    for(i=0;i<n;i++){
        cout<<"Enter the detail of "<<i+1<<"student"<<endl;
        ptr[i].getmarks();
    }
    for(i=0;i<n;i++){
        ptr[i].display();N
```

```

    }
    delete[]ptr;
    return 0;
}

```

4. Write a program to add two complex numbers by using dynamic constructor

```

#include<iostream>
using namespace std;
class complex{
    private:
        int *real,*imag;
    public:
        complex(){
            real= new int;
            imag= new int;
            *real=0;
            *imag=0;
        }
        complex(int r,int i){
            real=new int;
            *real=r;
            imag=new int;
            *imag=i;
        }
        void display(){
            cout<<*real<<"+"<<*imag<<"i"<<endl;
        }
        void addcomplex(complex c1,complex c2){
            *real=*c1.real+*c2.real;
            *imag=*c1.imag+*c2.imag;
        }
        ~complex(){
            delete real;
            delete imag;
        }
};

int main(){
    complex c1(5,10);
    complex c2(5,10);
    complex c3;
    c3.addcomplex(c1,c2);
    c3.display();
    return 0;
}

```

5. Using class write a program that receives inputs principle amount, time and rate. Keeping rate 8% as the default argument, calculate simple interest for three customers .

```
#include<iostream>
using namespace std;
class bank{
    private:
        float pamount;
        int principle,rate,time;
    public:
        void set(float p,float t,float r=8){
            principle=p;
            time=t;
            rate=r;
        }
        void calculate(){
            pamount=(principle*time*rate)/100;
        }
        void display(){
            cout<<"Simple Intrest="<<pamount<<endl;
        }
};
int main(){
    int i,n;
    int principle,time;
    bank b[i];
    cout<<"Enter the number of customer"<<endl;
    cin>>n;
    for(i=0;i<n;i++){
        cout<<"Enter the principle,time"<<endl;
        cin>>principle>>time;
        b[i].set(principle,time);
        b[i].calculate();
        b[i].bdisplay()
    };
    return 0;
}
```

6. WAP in C++ to calculate simple interest from given principal, time and rate. Set the rate to 15 % as default argument when rate is not provided and also implement the concept of dynamic initialization of object.

```
#include<iostream>
using namespace std;
class bank{
    private:
        float pamount;
        float principal,rate,time;
    public:
        void set(float p,float t,float r=15){
            principal=p;
            time=t;
            rate=r;
        }
        void calculate(){
            pamount=(principal*time*rate)/100;
        }
        void display(){
            cout<<"Simple Intrest="<<pamount<<endl;
        }
};
int main(){
    float principal,time,rate;
    bank b1;
    cout<<"Enter the principal,time,rate"<<endl;
    cin>>principal>>time>>rate;
    if(rate!=0){
        b1.set(principal,time,rate);
    }
    else{
        b1.set(principal,time);
    }
    b1.calculate();
    b1.display();
    return 0;
}
```