```
//dynamic allocation 1
#include<iostream>
using namespace std;

int main()
{
    //dynamically allocates memory
    int *i=new int{50};
   //new =is like malloc {allocating the space according to the data type}
  //memory leaks and dangling pointer
  float*f=new float{50.40f};
        cout << *i<<endl;
        cout << *f<<endl;
        //printing the variable values
        //deallocate the memory or free the memory for the  next usage
        // otherwise the data and the address will be left like that.
        delete i;
        i=nullptr;
        //for that set the pointer to null like nullptr
  delete f;
  f=nullptr;
        cout << *i<<endl;
        cout << *f<<endl;
        //memory allocation is Done at runtime
        return 0;
}
```

```
//dynamic allocation 2
#include <iostream>
using namespace std;

int main() {
```

```
   // dynamically allocate memory
   int* point_int = new int{45};//memory leaks and dangling pointers
         //new =variable; allocate memory for integer.
         // *point_int is a pointer (new) is the keyword which shows the location for alloction of the
memory
   float* point_float = new float{45.45f};


   cout << *point_int << endl;
   cout << *point_float << endl;


   // deallocate the memory or free the memory for the next usage
   // set pointers to nullptr
   delete point_int;
   delete point_float;


   cout<<*point_int<<endl;
   cout<<*point_float<<endl;
   //memory allocation is DONE at runtime
   return 0;
}
```

```
//dynamic allocation 3
// C++ program to store Rank of n number of students and display it
// where n is the number of students entered by the user

#include <iostream>
using namespace std;

int main() {

   int strength;
```

```
   cout << "Enter total number of students_strength: ";
   cin >> strength;
   float* ptr;

   //dynamic  & static memory allocation

   // memory allocation of num number of floats
   ptr = new float[strength]; //block of memory is allocated in bytes like(malloc)

   cout << "Enter the Rank of students." << endl;
   for (int i = 0; i < strength; ++i) {
   cout << "Student" << i + 1 << ": ";
   cin >> *(ptr + i);//takes user input and stores it in the memory location pointed to by (ptr + i)
//  ptr + i: Moves the pointer ptr to the ith element (via pointer arithmetic).
// *(ptr + i): Dereferences the pointer to access the memory at that location.
   }
   cout<<"size="<<sizeof(strength);  //New character =[sizeof] which will count the length of that

   cout << "\nDisplaying Rank of students." << endl;
   for (int i = 0; i < strength; ++i) {
   cout << "Student" << i + 1 << ": " << *(ptr + i) << endl;
   }
   // ptr memory is released
   delete[] ptr; //deleting the array which is pointing to this ptr {so *ptr ADDRESS will be deleted and becomes 0 for ptr}
   ptr = nullptr; //for avoiding the trash value
         //like linkedlist use the word NULL for the end of file
   cout<<"ptr= "<<ptr<<endl; //null (o) value
   cout<<*ptr<<endl;
   return 0;
}
```

```
//dynamic allocation 4
// C++ program to store GPA of n number of students and display it
// where n is the number of students entered by the user

#include <iostream>
using namespace std;

int main() {

   int num;
   cout << "Enter total number of students: ";
   cin >> num;
   float* ptr;

   //dynamic  & static memory allocation
   // memory allocation of num number of floats
   ptr = new float[num]; //block of memory is allocated in bytes like(malloc)

   cout << "Enter GPA of students." << endl;
   for (int i = 0; i < num; ++i) {
   cout << "Student" << i + 1 << ": ";
   cin >> *(ptr + i);
   }
   cout<<"size="<<sizeof(num);  //New character =[sizeof] which will count the length of that

   cout << "\nDisplaying GPA of students." << endl;
   for (int i = 0; i < num; ++i) {
   cout << "Student" << i + 1 << ": " << *(ptr + i) << endl;
   }
```

```
   // ptr memory is released
   delete[] ptr; //deleting the array which is pointing to this ptr {so *ptr ADDRESS will be deleted and becomes 0 for ptr}
   ptr = nullptr; //for avoiding the trash value
         //like linkedlist use the word NULL for the end of file
   cout<<"ptr= "<<ptr<<endl; //null (o) value
   cout<<*ptr<<endl;
   return 0;
}
```

```
//dynamic allocation 5
//Dynamic memory allocation using objects and constrcutors
#include <iostream>
using namespace std;
class men {
  int age;
  public:    // constructor initializes age to 12
   void Entry_Age(){    //Entry_age is the variable for the function
         cout<<"Enter the age"<<endl;
         cin>>age;
         }
   void Display_Age() {  //Diaplay_age is the variable for the function
      cout << "Age = " << age << endl;
   }
};
int main() { //and the [ NEW ] is used in the main only for allocating the memory
         men m1;  //creating the m1 variable
         // dynamically declare men object
   men *ptr = new men(); //men::men() have to understand once we learn Constrcutors
   ptr->Entry_Age();    //writing format for entering the age
   // call getAge() function
   ptr->Display_Age();  //writing format for displaying the age
```

```
   // ptr memory is released
   delete ptr;
   return 0;
}
```

```
//dynamic allocation 6
//Dynamic memory allocation using objects and constrcutors
#include <iostream>
using namespace std;
class Student {
  int age;
  public:    // constructor initializes age to 12
   void setAge(){
            cout<<"Enter the age"<<endl;
            cin>>age;
            }
   void getAge() {
      cout << "Age = " << age << endl;
   }
};
int main() { //and the [ NEW ] is used in the main only
            Student s1;
            // dynamically declare Student object
   Student *ptr = new Student(); //Student::Student() have to understand once we learn
Constrcutors
   ptr->setAge();
   // call getAge() function
   ptr->getAge();
   // ptr memory is released
   delete ptr;
   return 0;
}
```