EX.NO : 2(A)	C++ DYNAMIC MEMORY ALLOCATION
DATE :	

PROGRAM STATEMENT :

To Write A CPP Program to allocate memory dynamically for an integer variable. (Note: p_var = new typename;)

ALGORITHM :

1. Start
2. Define a class `var_space` with a member function `allocateSpace()`.
3. Inside `allocateSpace()`, dynamically allocate memory for an integer using `new int`.
4. Read an integer value from the user and store it in the dynamically allocated memory.
5. Display the stored integer value and End.

PROGRAM :

```
#include <iostream>
using namespace std;
class var_space
{
 public:
 void allocateSpace()
 {
      int *p_var=new int;
      cin>>*p_var;
      cout<<"Integer Value is : "<<*p_var;

 }
};
int main()
{
   var_space s;
   s.allocateSpace();
}
```

OUTPUT :

	Input	Expected	Got	
✔	10	Integer Value is : 10	Integer Value is : 10	✔
✔	20	Integer Value is : 20	Integer Value is : 20	✔

Passed all tests! ✔

RESULT:

Thus, the C++ program to allocate memory dynamically for a an integer variable is created successfully.

EX.NO : 2(B)	C++ STATIC CLASS MEMBERS
DATE :	

PROGRAM STATEMENT :

To Write A CPP Program to create class SquareBox and calculate the volume of the SquareBox, make use of static member variable and static member function in the class SquareBox.

ALGORITHM :

1. Start
2. Define a class SquareBox (in your code it's Square) with:
3. Static member variables objcount and stagecount.
4. Constructor to initialize dimensions and increment static counters.
5. A member function Volume() to calculate volume.
6. Create two SquareBox objects using the input dimensions, and display their volumes.
7. Print the total number of objects created and the final stage count, then End.

PROGRAM :

```
#include <iostream>
using namespace std;
class Square {
   public:
   static int objcount,stagecount;
     Square(double l,double b,double h)
     {
       length=l;
       breadth=b;
       height=h;
       cout<<"Constructor called."<<endl;
       objcount++;
       stagecount++;
     }
     double Volume()
     {
         return length*breadth*height;
     }

  private:
     double length;
     double breadth;
     double height;
};
```

```
int Square::objcount=0;
int Square::stagecount=0;
int main(void)
{
   int l,b,h,l1,b1,h1;
   cin>>l>>b>>h>>l1>>b1>>h1;
   cout<<"Inital Stage Count: "<<Square::stagecount<<endl;
   Square s1(l,b,h);
   cout<<"Volume :"<<s1.Volume()<<endl;
   Square s2(l1,b1,h1);
   cout<<"Volume :"<<s2.Volume()<<endl;
   cout<<"Total objects: "<<Square::objcount<<endl;
   cout<<"Inital Stage Count: "<<Square::stagecount<<endl;

  return 0;
}
```

OUTPUT :

	Input	Expected	Got	
✔	9 8 4 1 6 7	Inital Stage Count: 0 Constructor called. Volume :288 Constructor called. Volume :42 Total objects: 2 Inital Stage Count: 2	Inital Stage Count: 0 Constructor called. Volume :288 Constructor called. Volume :42 Total objects: 2 Inital Stage Count: 2	✔
✔	12 22 33 44 55 66	Inital Stage Count: 0 Constructor called. Volume :8712 Constructor called. Volume :159720 Total objects: 2 Inital Stage Count: 2	Inital Stage Count: 0 Constructor called. Volume :8712 Constructor called. Volume :159720 Total objects: 2 Inital Stage Count: 2	✔

RESULT:

Thus, the Write A CPP Program to create class SquareBox and calculate the volume of the SquareBox, make use of static member variable and static member function in the class SquareBox created successfully.

EX.NO : 2(C)	C++ FUNCTION OVERLOADING
DATE :	

PROGRAM STATEMENT :

To Write a CPP Program to overload a function to perform sum of two integers and sum of three integers.

ALGORITHM :

1. Define a class Sum with two overloaded methods named sum():
2. One method takes two integers and prints their sum.
3. The other method takes three integers and prints their sum.
4. In the main() function, declare an object s of class Sum and three integer variables (x, y, z).
5. Accept two integer inputs (x and y) from the user using cin.
6. Call the two-parameter sum(x, y) function to calculate and display the sum of two numbers.
7. Accept three integer inputs (x, y, z) and call the three-parameter sum(x, y, z) function to display the sum of three numbers.

PROGRAM :

```
#include<iostream>
using namespace std;
class Sum
{
   public:
   void sum(int x,int y)
   {
      cout<<"Sum of two Numbers="<<x+y<<endl;
   }
   void sum(int x,int y,int z)
   {
      cout<<"Sum of three Numbers="<<x+y+z;
   }
};
int main()
{
   Sum s;
   int x,y,z;
   cin>>x>>y;
   s.sum(x,y);
   cin>>x>>y>>z;
```

```
    s.sum(x,y,z);
    return 0;
}
```

OUTPUT :

	Test	Input	Expected	Got	
✔	1	10 20 10 20 30	Sum of two Numbers=30 Sum of three Numbers=60	Sum of two Numbers=30 Sum of three Numbers=60	✔
✔	2	100 200 100 200 300	Sum of two Numbers=300 Sum of three Numbers=600	Sum of two Numbers=300 Sum of three Numbers=600	✔
✔	3	23 56 56 75 23	Sum of two Numbers=79 Sum of three Numbers=154	Sum of two Numbers=79 Sum of three Numbers=154	✔

RESULT:

Thus, the C++ Program to overload a function to perform sum of two integers and sum of three integers variable is created successfully.

EX.NO : 2(D)	C++ OPERATOR OVERLOADING
DATE :	

PROGRAM STATEMENT :

To Write a CPP Program to overload the Operator (++) i.e. on invoking it the incrementation should happen by some random value.

ALGORITHM :

1. Define a class op with two public integer members a and b.
2. Overload the postfix ++ operator (operator++(int)) to add the value of b to a.
3. In the main() function, create an object o of class op.
4. Read two integer inputs from the user (o.a and o.b) using cin.
5. Use the overloaded o++ operator to perform a = a + b, then output the result stored in o.a.

PROGRAM :

```
#include<iostream>
using namespace std;
class op{
   public:
   int a,b;
   void operator ++(int){
      a=a+b;
   }
};
int main(){
   op o;
   cin>>o.a>>o.b;
   o++;
   cout<<o.a;
}
```

OUTPUT :

	Test	Input	Expected	Got	
✔	1	25 25	50	50	✔
✔	2	10 45	55	55	✔
✔	3	45 23	68	68	✔

RESULT:

Thus, Write a CPP Program to overload the Operator (++) i.e. on invoking it the incrementation should happen by some random value is created successfully.