SP22-BCS-083 (B)

Aqsa Hussain

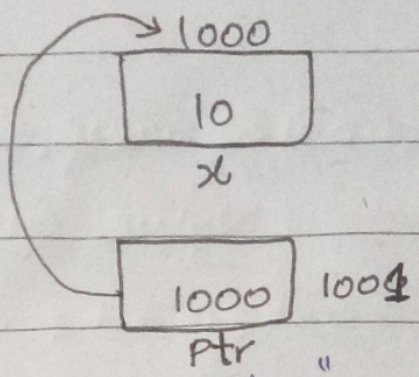
Date:

Assignment No: 2

DSA (Lab)

Program no: 1

```
int main( )
① int x=10;
② int *ptr= &x;
```




```
③ cout << "program 1: Basic pointer"<<endl;
```

Program 1: Basic Pointer

```
④ cout << "value of x:" << x <<endl;
```

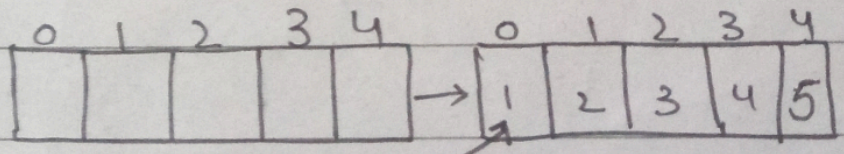
value of x: 10

```
⑤ cout << " value pointed to by ptr:" <<*ptr<<
                                          endl;
```

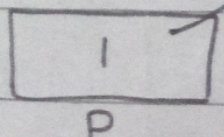
value pointed to by ptr: 10

Program no: 2:-

```
int main( )
① int arr[ ]={1, 2, 3, 4, 5};
```




```
② int *p = arr;
```



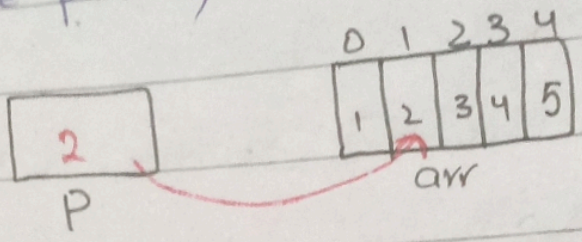
③ cout << "In program 2 : pointer Arithematic" << endl;

pointer Arithematic

④ cout << "value at p:" << *p << endl;

value at p : 1

⑤ ~~cout~~ p++;



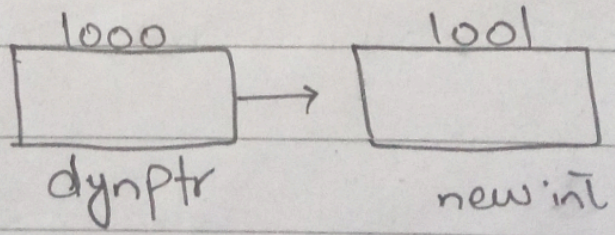
⑥ cout << " value at p++ : " << *p << endl

value at p++ : 2

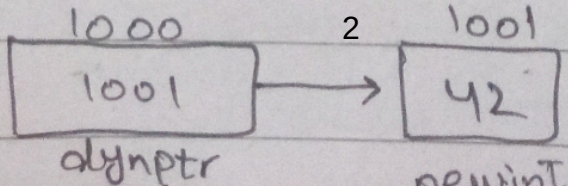
Program no : 3 :-

int main ()

① int *dynptr = new int ;



② *dynptr = 42;



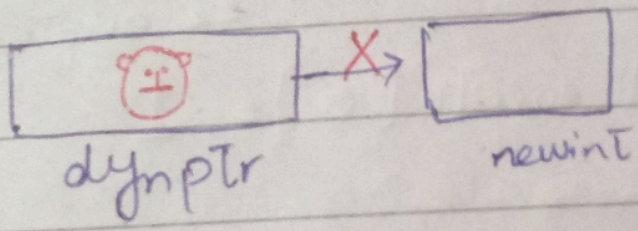
③ cout << "\nprogram 3: Dynamic Memory Allocation" << endl;

Dynamic Memory Allocation

④ cout << "value pointed to by dynptr:" << *dynptr << endl;

value pointed to by dynptr: 42

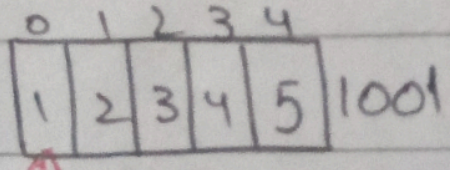
⑤ delete dynptr;



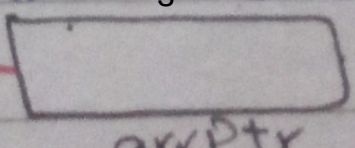
Program no : 4:-

① int main

② int numbers[ ] = {1, 2, 3, 4, 5};



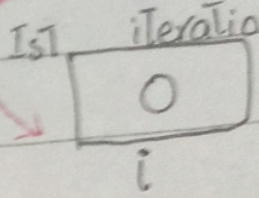
③ int* arrptr = numbers;



④ cout << "In program 4: pointer to Array" << endl;

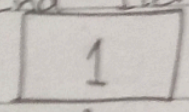
Program 4: pointer to Array:

⑤ for(int i=0; i<5; i++) {

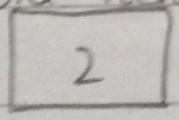


⑥ cout << " value at arrptr[" << i << "]: " << arrptr[i] << endl

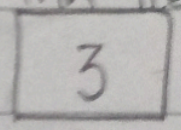
value at arrptr[0] = 1



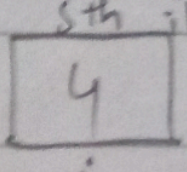
value at arrptr[1] = 2



value at arrptr[2] = 3



value at arrptr[3] = 4



Day: Date:

value at arrptr[4] = 5

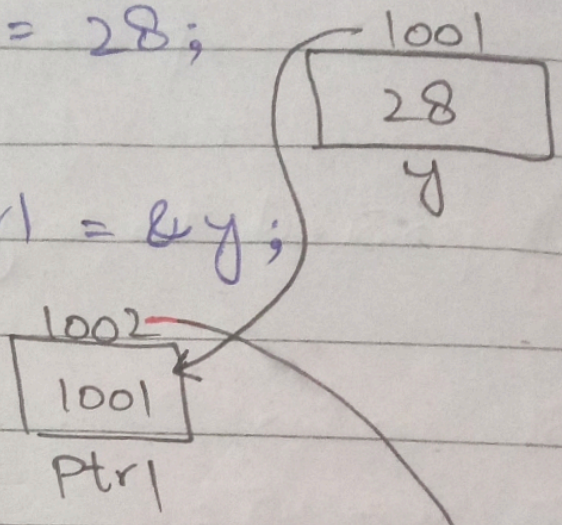
}

---

program no: 5

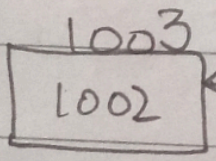
① int main()

② int y = 28;



③ int *ptr1 = &y;

④ int **ptr2 = &ptr1;



⑤ cout << "-In program 5 : pointer to pointer"
<< endl;

program 5 : pointer to pointer

⑥ cout << " value of y: " << y << endl;

value of y: 28

Day: Date:

⑦ cout << "value pointed to by ptr1: " ~~<< endl~~ << *ptr1 << endl

value pointed to by ptr1: 28

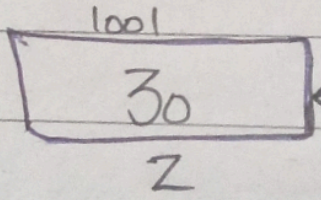
⑧ cout << "value pointed to by ~~ptr2~~ ptr2(ptr1): " << **ptr2 << endl

value of pointed to by ptr2(ptr1) = 28

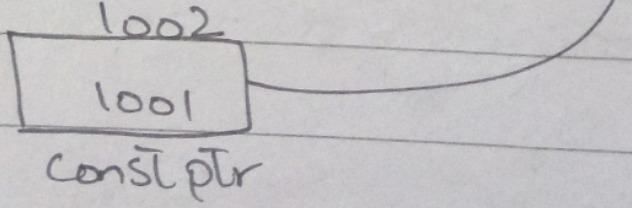
Program no: 6

① int main()

② const int z = 30;



③ const int * const ptr = &z;



④ cout << "In program 6: pointer to constant" << endl;

Day: Date:

program 6: pointer to constant.

⑤ cout << "value of z:" << z << endl;

value of z : 30

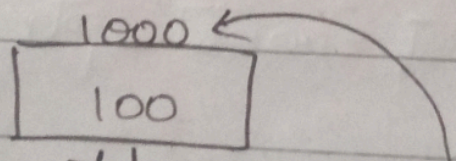
⑥ cout << "value of pointed to by consiptr:" << ~~const~~ siptr << end l;

~~cout <<~~ "value of pointed to by consiptr: 30

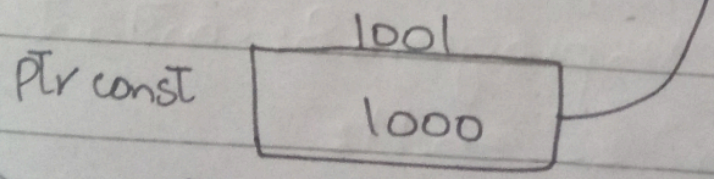
Program no: 7

① int main ( )

② int const x1 = 100;



③ int const *ptr_const = & x1;



④ cout << "In program 7: pointer to constant ~~pada~~ " << end l;

~~Cout~~ Program 7: pointer to constant data:

Day: Date:

(5) cout << " value of x1: " << x1 << endl;

value of x1: 100

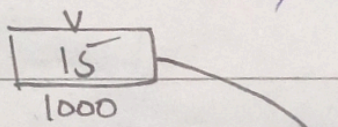
(6) cout << "value pointed to by ptr_const << *ptr_const << endl

value pointed to by ptr-const: 100

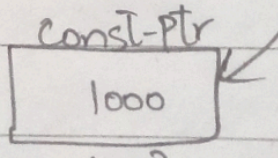
Program no: 8:-

① int main()

② int v = 15;



③ int * const const_ptr = &v;



④ cout << "In program 8: Constant pointer" << endl;

program8: Constant pointer

⑤ cout << value of v:" << v << endl;

value of v = 15

Day: Date:

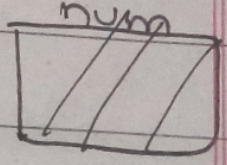
⑥ cout << "Value pointed to by constant-ptr: " << const_ptr << endl;
value pointed to by const_ptr: 15.

Program no: 9:-

int main()

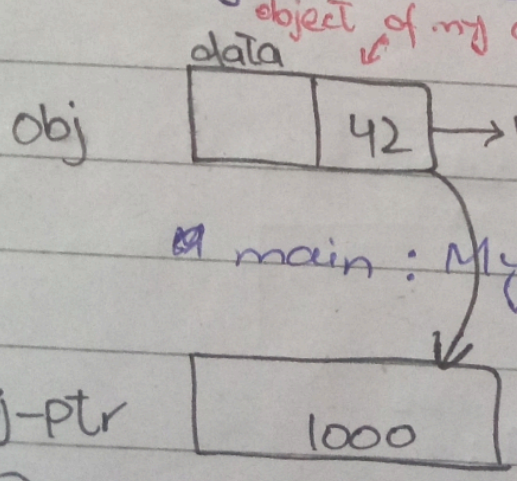
① class My class {
public:
int data;

② My class (int val): data(val) { }
};



③ Start → My class obj (42);

④ My class * obj_ptr = & Obj;



⑤ cout << "In program 9: pointer to objects" << endl;

Program 9: pointer to objects:

Date:

Day:

⑥ cout << " value of obj.data: " << obj.data << endl;

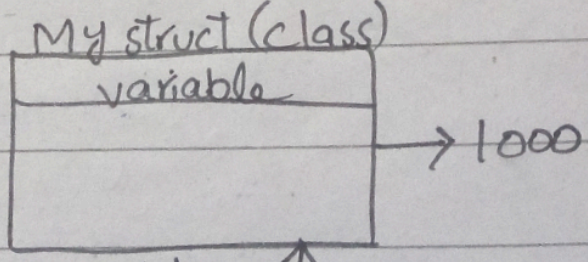
~~cout <~~ Value of obj.data: 42

⑦ cout << " value of pointed to by obj_ptr->data: " << obj_ptr->data << endl;

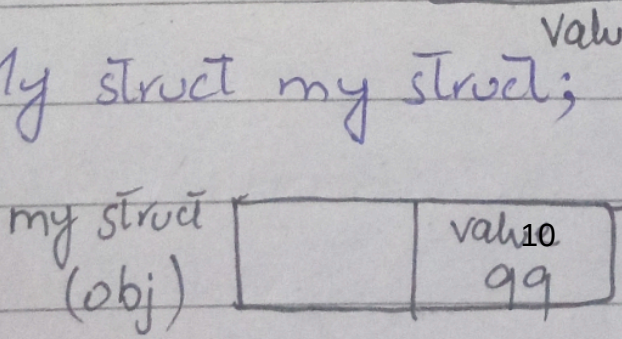
value of pointed to by obj_ptr->data: 42:

Program no: 10:-

```
① int main()
② class My Struct
③ public:
④     int value;
⑤ };
```

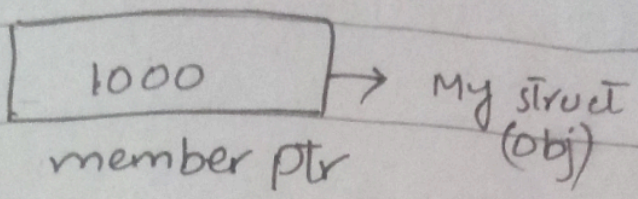


⑥ My struct my struct;



Day: Date:

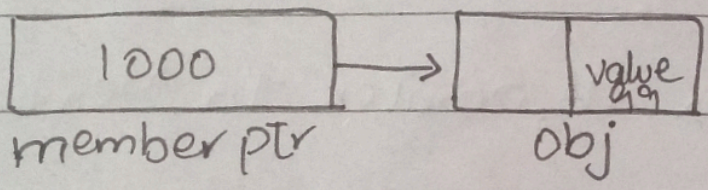
① int MyStruct :: *memberptr = &MyStruct :: value;



② cout << "In Program 10: Pointer to member variable" << endl;

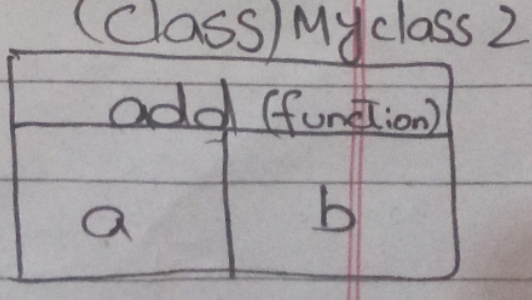
Program 10: Pointer to member variable.

③ ~~cout <<~~ mystruct.*memberptr = 99;

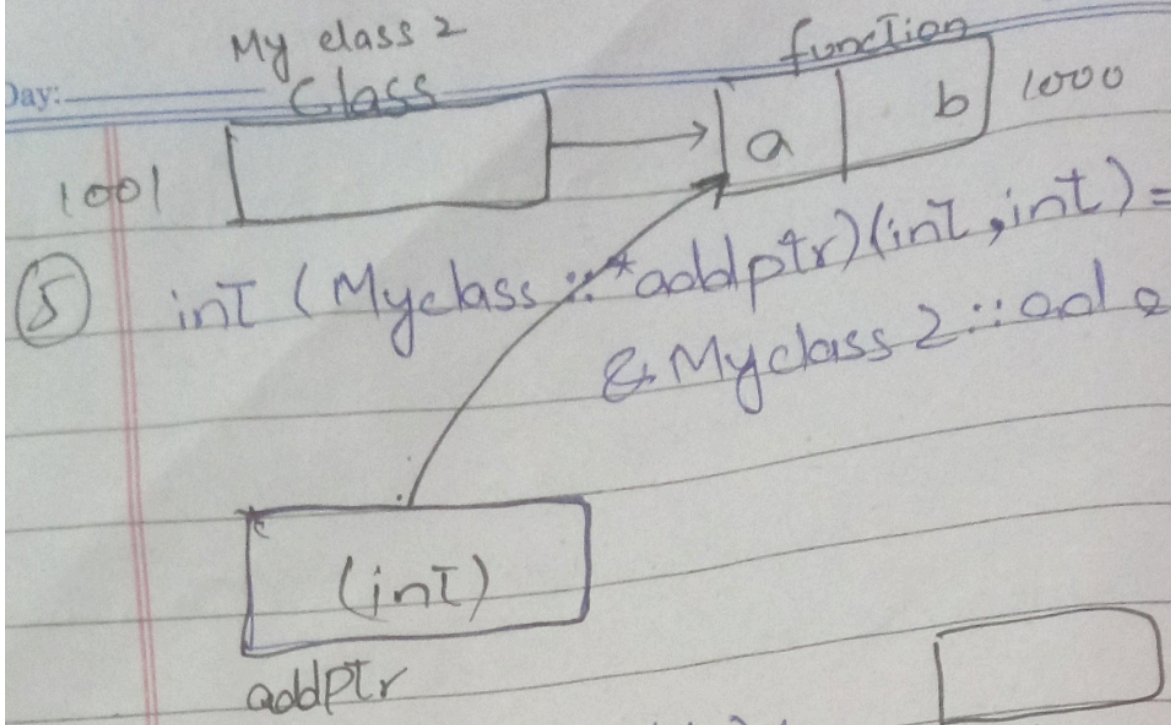


④ cout << "value of mystruct.value:"
<< mystruct.value << endl;
value of mystruct.value = 99;

Program no : 11:-



① int main()
② class Myclass 2 {
③ public:
④ int add(int a, int b) { return a+b; }



(5) int (Myclass::*addptr)(int, int) = &Myclass2::add;

(6) My class 2 obj2;

(7) cout << "In program 11: pointer to Member Function" << endl;

Program 11 pointer to Member Function.

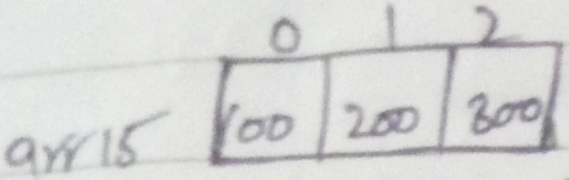
(8) cout << "Result of obj2.add(3,4) using addptr:" << (obj2.*addptr)(3,4) << endl;

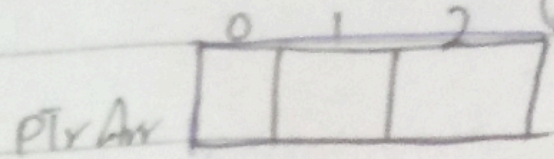
9 Result of obj2.add(3,4) using addptr: 7

Program no: 12:-

(1) int main()

(2) int arr[5] = {100, 200, 300};



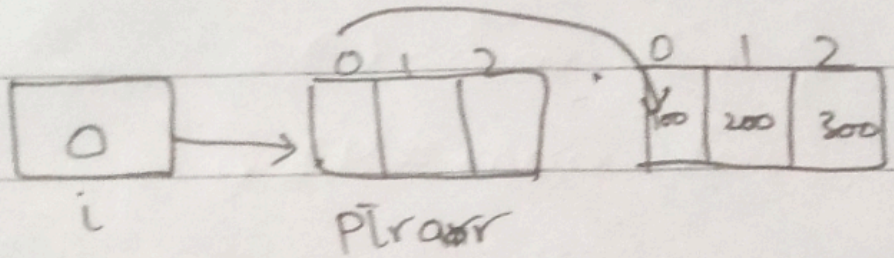


③ int * ptrArr[3];

④ for(int i=0; i<3; i++) {

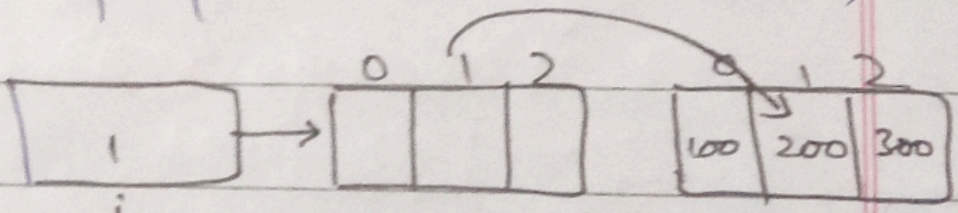
⑤ ptrArr[i] = &arr15[i];

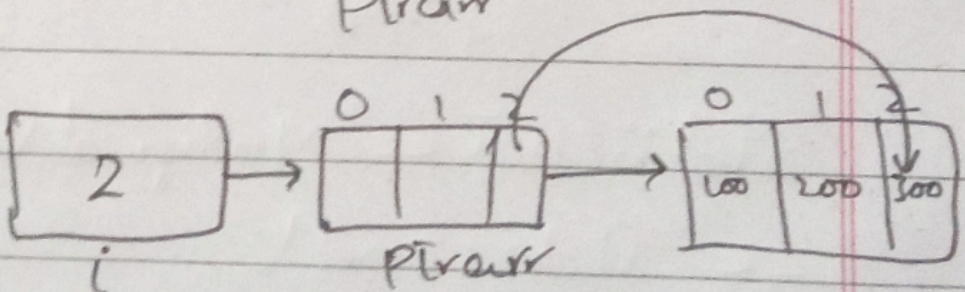
}



~~ptrArr[i] =~~

⑥ cout << "\n program 12: pointer to Array of pointers " << endl;





⑦ for(int i=0; i<3; i++) {

⑧ cout << "value pointed to by ptrarr [" << i << "]: " << *ptrarr[i] << endl;

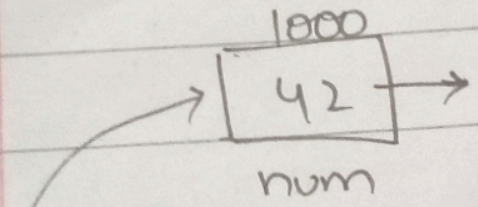
pointed to by ptrarr 0: 100.

Day: Date:

pointer to by ptrarr[1] = 200.
pointer to by ptrarr[2] = 300

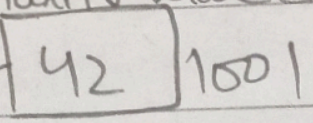
Program no: 13:-

```
① void modifyValue (int *ptr)
② {  (*ptr)++;
③ } int main()
④ { int num = 42;
```




```
⑤ modifyvalue(&num);
```

modify value(int*)



ptr

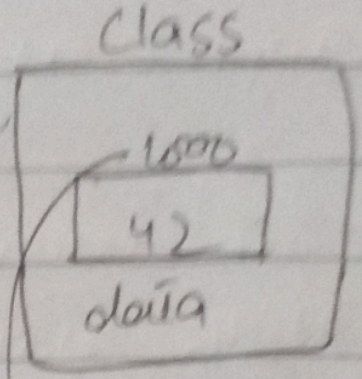
```
⑥ Std::cout << "Modified value:" << num
   << std::endl;
```

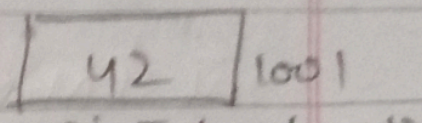
Modified value : 43

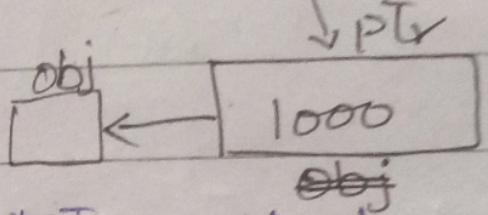
Program no: 14:-

```
class Myclass {
Public:
int data=42;
void printData() {
Std:: cout << " Data: "
     << ~~endl~~ data<<
std:: endl;

Data: 42
int main()
My class obj:
int My class ::*ptr=& My class::data;
(obj.*ptr)=84;
obj.printdata();
```



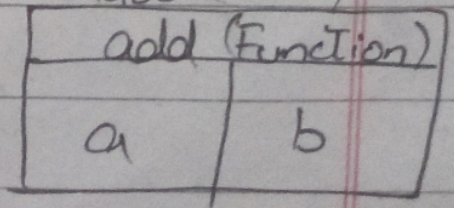




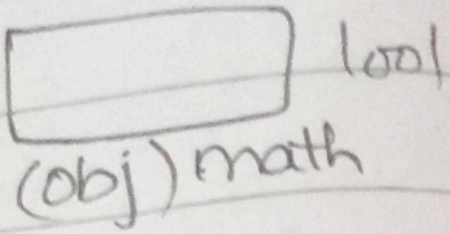
Data 84:

Program no: 15:-

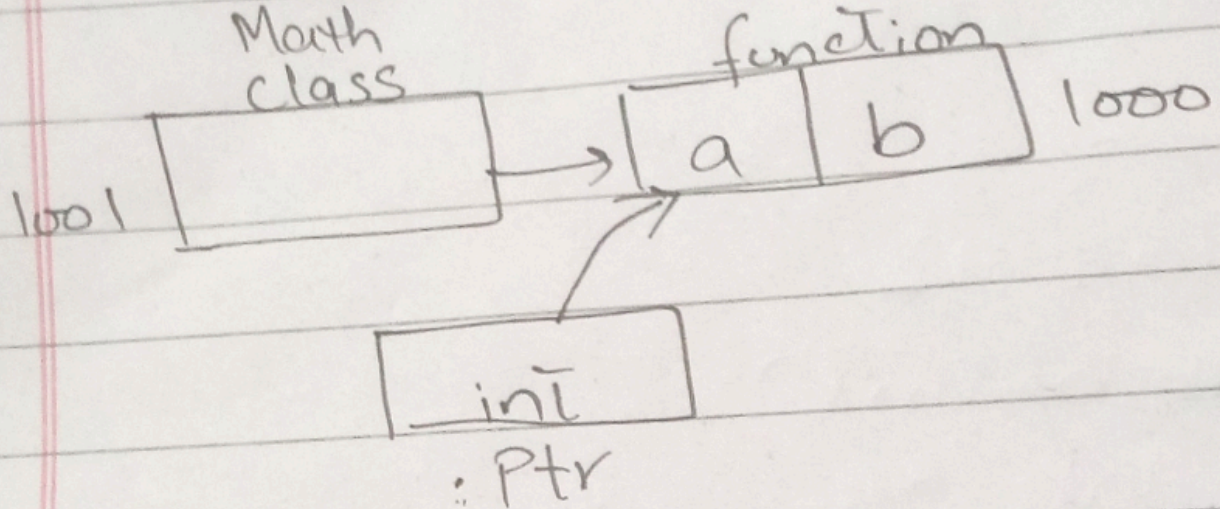
```
class Math{
Public:
int add (int a, int b){
```



```
    return a+b;
  }
};  int main() {
    Math math;
```




```
int (Math ::*ptr)(int,int)=
        &Math::add;
```




```
std:: cout << sum << (math.*ptr)
  (3, 4) << std:: endl;
return 0;
```

Sum : 7