Name-Rohan Ghosh

Batch- BCS2B

ID - 181001001122

GROUP-A

Choose the correct one:-

I. A constructor is called whenever

ANS: a) an object is declared.

II. Which type of class has only one unique value for all the objects of that same class?

ANS:c. static.

III. intmain(){

int x=10; int&p=x;

cout<<&p<<&x;

}

ANS: c. print the address of x twice.

IV. Choose the correct statement

ANS: c) struct is by default public class is by default private.

V. C++ was originally developed by

ANS: c) Bjarne Strousstup .

GROUP-B

1 Is it necessary to accept a reference in the copy constructor?

ANS: YES,it necessary to accept a reference in the copy constructor

2 Compare pointer and reference variable?

ANS:

POINTER	Reference variable
A pointer can be re-assigned.	Whereas, a reference cannot,

	and must be assigned at initialization.
A pointer has its own memory address and size on the stack.	Whereas a reference shares the same memory address (with the original variable) but also takes up some space on the stack.
Pointer can be assigned NULL directly.	Whereas reference cannot assign NULL directly.
We can have pointers to pointers offering extra levels of indirection.	Whereas references only offer one level of indirection.
Various arithmetic operations can be performed on pointers.	Whereas reference we can't perform arithmetic operations.

3 What are the applications of scope resolution operator in C++?

ANS:1) To access a global variable when there is a local variable with same name.

2) To define a function outside a class.

3) To access a class's static variables.

4) In case of multiple Inheritance.

5) For namespace.

4 Write the special characteristics of friend function.

ANS:1) The function is not in the scope of the class to which it has been declared as a friend.

2) It cannot be called using the object as it is not in the scope of that class.

3) It can be invoked like a normal function without using the object.

4) It cannot access the member names directly and has to use an

object name and dot membership operator with the member name.

5 Write the advantages and disadvantages of inline function.

ANS:Advantages of Inline function are follows:

1) Function call overhead doesn’t occur.
3) It also saves overhead of a return call from a function.
4) When you inline a function, you may enable compiler to perform context specific optimization on the body of function. Such optimizations are not possible for normal function calls. Other optimizations can be obtained by considering the flows of calling context and the called context.
5) Inline function may be useful (if it is small) for embedded systems because inline can yield less code than the function call preamble and return.

Disadvantages of Inline function are follows:

1) If you use too many inline functions then the **size of the binary executable file will be large**, because of the duplication of same code.

2) Too much inlining can also reduce your instruction cache hit rate, thus **reducing the speed** of instruction fetch from that of cache memory to that of primary memory.

3) Inline function may **increase compile time overhead** if someone changes the code inside the inline function then all the calling location has to be recompiled because compiler would require to replace all the code once again to reflect the changes, otherwise it will continue with old functionality.

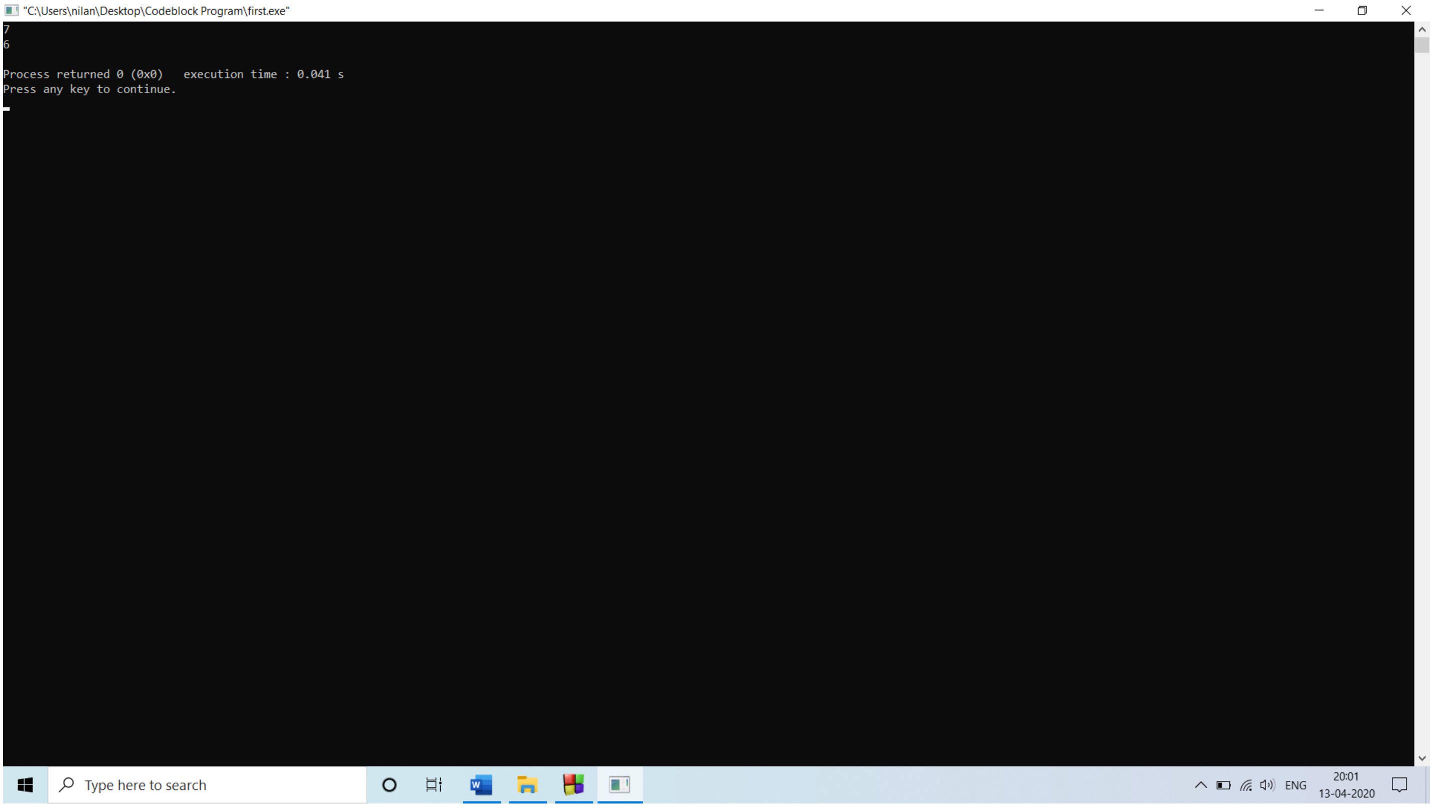
GROUP-C

6 What will be the output of the followings? – Explain.

ANS: For Source Code A

```
class test {
public:
static int n;
test (){
n++;
};
~test () {
n–;
}
};
int test::n=0;
int main (){
test a;
test b[5];
test * c = new test;
cout<<a.n<<endl;
delete c;
cout<<test::n <<endl;
return 0;
}
```

OUTPUT



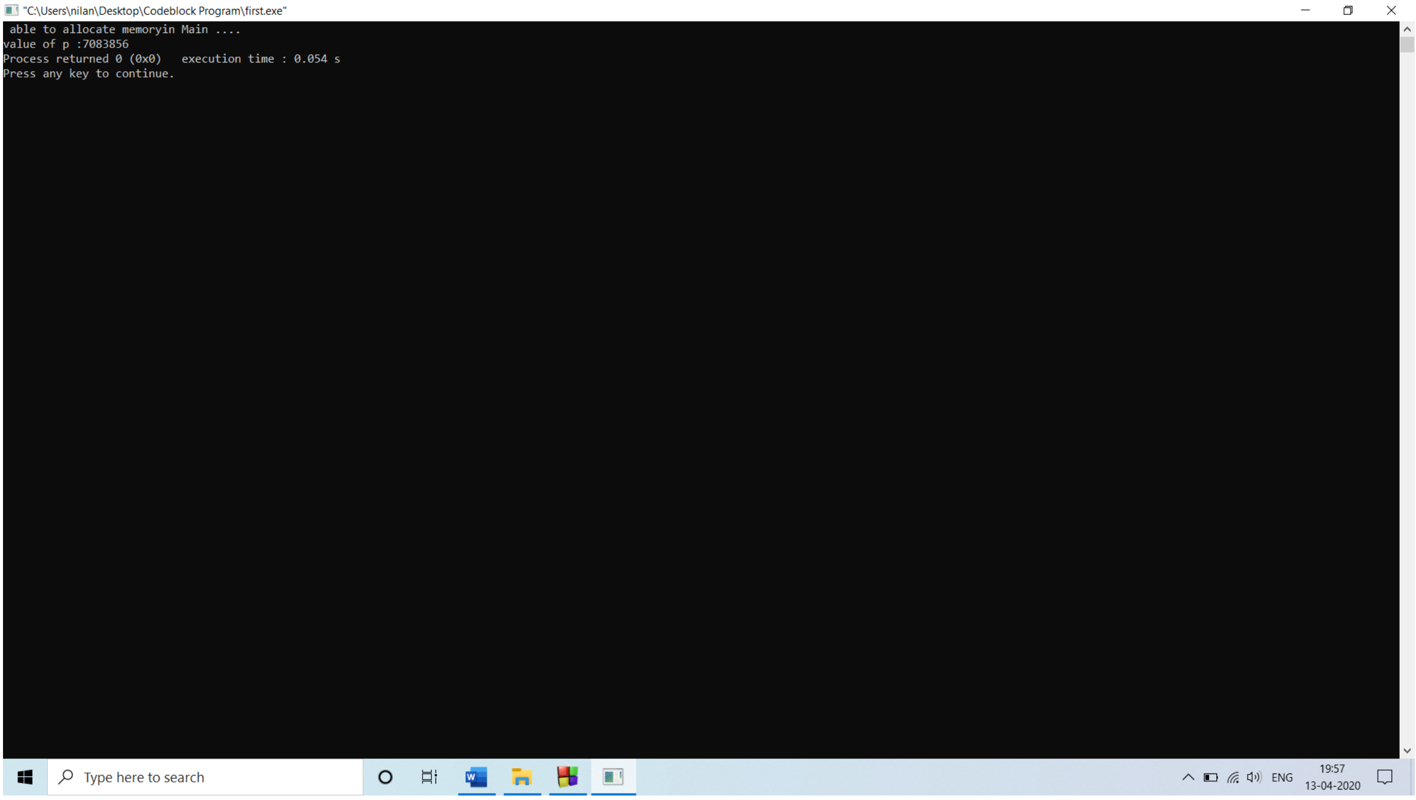
```
7
6

Process returned 0 (0x0)   execution time : 0.041 s
Press any key to continue.
```

Test a calls the default constructor n changes to 1, then b[5] calls the same constructor 5 times and then c is created does the same purpose. Final value thus becomes 7 which is first print. Now the ~test() perfroms its task and value changes to 7 - - which is 6. test::n thus prints that value of 6.

For Source Code B

```
#include<iostream>
#include<malloc.h>
using namespace std;
class A{
public:
int p;
A(){
p=4;
cout<<"constructor A"<<"\np value is:"<<p<<"\n";
}
};
int main(){
A *a= (A*)malloc(sizeof(A));
if(a== NULL)
cout<<" memory not allocated";
else
cout<< " able to allocate memory";
cout<<"in Main .... \nvalue of p :"<<a->p;
}
```

OUTPUT



"C:\Users\nilan\Desktop\Codeblock Program\first.exe"

```
 able to allocate memoryin Main ....
value of p :7083856
Process returned 0 (0x0)   execution time : 0.054 s
Press any key to continue.
```

A *a creates a pointer to the constructor with memory allocation of the function size itself. The constructor is never invoked so p=4 is never printed. But a valid memory allocation is done, so its not NULL hence it goes in the else section and a->p refers to the memory allocation reference of the value and thus prints the value of 7083856.