Question 1: Inline functions are useful when

A. Function is large with many nested loops
B. Function has many static variables
C. Function is small and we want to avoid function call overhead.
D. None of the above

Question 2

```
#include<iostream>
using namespace std;
int x = 1;
void fun()
{
   int x = 2;
   {
      int x = 3;
      cout << ::x << endl;
   }
}
int main()
{
   fun();
   return 0;
}
```

A. 1
B. 2
C. 3
D. 0

Question 3

Predict the output of following C++ program

```
#include<iostream>
using namespace std;

union A {
  int a;
  unsigned int b;
  A() { a = 10; }
  unsigned int getb() {return b;}
};

int main()
{
   A obj;
   cout << obj.getb();
   return 0;
}
```

A. Compiler Error: union can't have functions
B. Compiler Error: can't access private members of A
C. 10
D. garbage value

Question 4: Which of the following is true about inline functions and macros.

A. Inline functions do type checking for parameters, macros don't
B. Macros are processed by pre-processor and inline functions are processed in later stages of compilation.
C. Macros cannot have return statement, inline functions can.
D. Macros are prone to bugs and errors, inline functions are not.
E. All of the above

Question 5: How can we make a C++ class such that objects of it can only be created using new operator? If user tries to create an object directly, the program produces compiler error.

A. Not possible
B. By making destructor private
C. By making constructor private
D. By making both constructor and destructor private

Question 6: Would destructor be called, if yes, then due to which vector?

```
#include <iostream>

#include <vector>

using namespace std;


class a

{

public :

   ~a()

   {

      cout << "destroy";

   }

};
```

```
int main()
{
  vector <a*> *v1  = new vector<a*>;
  vector <a> *v2  = new vector<a>;
  return 0;
}
```

A. v1
B. v2
C. v1 and v2
D. no destructor call

Question 7

```
#include<iostream>
using namespace std;

int x[100];
int main()
{
   cout << x[99] << endl;
}
```

A. Unpredictable
B. Runtime error
C. 0
D. 99

Question 8

```
#include<iostream>
using namespace std;
int main ()
{
    int cin;
    cin >> cin;
    cout << "cin" << cin;
    return 0;
}
```

A. error in using cin keyword
B. cin+junk value
C. cin+input
D. Runtime error

Question 9: The associativity of which of the following operators is Left to Right, in C++ ?

A. Unary Operator
B. Logical not
C. Array element access
D. addressof

Question 10: A member function can always access the data in _____ , (in C++).

A. the class of which it is member
B. the object of which it is a member
C. the public part of its class
D. the private part of its class