1. Which of the following decides the behaviour of the object?

Answers

1. data member

2. member function\*

3. data member and member function both

4. none of the above

2. ...........are member functions of the class,which modifies the state of the object.

Answers

1. constructor

2. Inspectors

3. Iterator

4. Mutators\*

5. Destructor

3. What is the use of Namespace?

Answers

1. To encapsulate the data

2. To structure a program into logical units\*

3. Encapsulate the data & structure a program into logical units

4. None of the mentioned

5. to combine data and functions together

4. Which of the following statement is incorrect about structure in c++?

Answers

1. we can encapsulate data member as well as member functions inside structure in c++

2. use of struct keyword,while declaring structure variable is optional

3. Default access specifier for members inside structure in c++ is private\*

4. none of the above

5. In which of the following cases inline functions may not work?

Answers

1. If the function has static variables and loops.

2. If the function has global and register variables.

3. If the function is recursive

4. both 1 and 3\*

5. both 1 and 2

6. which members get space inside object?

Answers

1. Non static data member

2. member functions

3. static as well as non-static data members

4. Both 1 and 2

5. Only 1\*

7. What is the output of this program?

#include <iostream>

using namespace std;

int func(int m = 10, int n)

{

int c;

c = m + n;

return c;

}

int main()

{

cout << func(5);

return 0;

}

Answers

1. 10

2. 15

3. Compile time error\*

4. Run time error

5. None of the above

8. which of the following statement is correct in context of class?

Answers

1. class is template for an object

2. Class is logical entity and it does not get space inside memory.

3. class is group of objects of same type i.e having same state and behaviour.

4. By defining class we are achieving encapsulation

5. 1, 2 and 4\*

9. #include <iostream>

#include <string>

using namespace std;

namespace A{

int var = 10;

}

namespace B{

int var = 5;

}

int main()

{ int var = 20;

using namespace B;

cout<<var;

}

Answers

1. 5

2. 10

3. 20\*

4. Error

5. None of the above

10. Which of the following permits function overloading in c++?

Answers

1. Argument type

2. Number of arguments

3. Sequence of arguments

4. All of these\*

5. None of these