

```
1. #include<iostream>
using namespace std;
void Sum(int n1, int n2 , int n3)
{
    cout<<"Sum="<< n1+ n2 + n3;
}
namespace NSum
{
    int Sum(int n1, int n3 , int n2)
    {
        return n1+ n2 + n3;
    }
}
int main()
{
    Sum(10, 20, 30);
    return 0;
}
```

Answers

1. Compile time error

2. Sum=60

3. Run time error

4. No output as we not catch output value

2. ....are member functions of the class, which is used to read the state of the object.

Answers

1. constructor

2. Mutators

3. Iterator

4. Inspectors

5. Destructor

3. What is output of this program?

```
class date
{
    int day,month,year;

    date()
    {
        day=25;
        month=4;
        year=2020;
    }

public:
    void printDate()
    {
        cout<<this->day<<"-"<<this->month<<"-"<<this->year<<endl;
    }
};

Int main()
{
    date today;
    today.printDate();
    return 0;
}
```

Answers

1. Garbage value
2. 25-4-2020
3. Runtime error

4. Compile time Error

```
4. #include<iostream>
using namespace std;
namespace NOuter
{
    int num1=100;
    int num2=200;
    namespace NInner
    {
        int num1=300;
        int num3=200;
    }
}
```

what will be the correct way to print num3 value?

Answers

1. using namespace NOuter::NInner; int main() {  
cout<<"num3="<<num3<<endl; return 0; }
2. using namespace NOuter; using namespace NInner; int main() {  
cout<<"num3="<<num3<<endl; return 0; }

3. Both options A and B are correct

4. Compile Time Error

```

5. #include<iostream>
using namespace std;
#pragma pack(1)
namespace NPoint
{
    class Point
    {
    };
    struct point
    {
    };
}
using namespace NPoint;
int main()
{
    cout<<"size of Point = "<<sizeof(Point)<<endl;
    cout<<"size of point = "<<sizeof(point)<<endl;
    return 0;
}
Answers

```

1. size of Point = 1      size of point = 1
2. size of Point = 0      size of point = 0
3. compile time error
4. run time error

```

6. #include<iostream>
using namespace std;
namespace NPoint
{
    class Point
    {
    };
}

```

if we create an object of Point class then the state of the object is -----and behavior of the object is -----.

Answers

1. nothing , nothing
2. compile time error, run time error
3. run time error, compile time error
4. garbage value, unexpected

7. When we can not declare a function inline?

Answers

1. If the function consists of loops.
2. If function consists of recursion functions.
3. If function consists of jump statements.
4. All of above

```

8. 1 #include<iostream>
2 using namespace std;
3 #pragma pack(1)
4 namespace NPoint
5 {
6     class Point
7     {
8         int xPosition;
9         int yPosition;
10        void Print()
11        {
12            cout<<"xPosition="<<xPosition<<endl;
13            cout<<"yPosition="<<yPosition<<endl;
14        }
15    };
16    struct point
17    {
18        int xPosition;
19        int yPosition;
20        void Print()
21        {
22            cout<<"xPosition="<<xPosition<<endl;
23            cout<<"yPosition="<<yPosition<<endl;
24        }
25    };
26 }
27 using namespace NPoint;
28 int main()
29 {
30     Point p1;
31     p1.xPosition=100;
32     p1.yPosition=200;
33     p1.Print();
34
35     point p2;
36     p2.xPosition=100;
37     p2.yPosition=200;
38     p2.Print();
39     return 0;
40 }

```

#### Answers

1. error on line no 30 31 32 33

2. error on line no 31 32 33

3. error on line no 35 36 37 38 35 36 37 38

4. error on line no 31 32 33 36 37 38

9. What is output of this program?

```
void fun(int=10,int=20,int=30);
void fun(int, int);

int main()
{
    fun(1,2);
    return 0;
}

void fun(int x,int y,int z)
{
    Cout<<endl<<x<<endl<<y<<endl<<z;
}

void fun(int x,int y)
{
    cout<<endl<<x<<endl<<y;
}
```

Answers

1. 1 2

2. Compile time error

3. 1 2 30

4. none of the above

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

Answers

1. Type of arguments
2. Number of arguments

3. Type, Number and Sequence of arguments

4. Return type of function

5. Both C and D

----- Compiled by Utkarsh Singh -----

**Attention:** This material is strictly confidential and intended solely for your use. It is imperative that you do not disseminate or share this material with the public, under any circumstances. Any breach of confidentiality will result in severe consequences. Protect the integrity of this material and the trust placed in you.