
1- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace MyNamespace {
    int x = 5;
}

int main() {
    using namespace MyNamespace;
    std::cout << "Value of x: " << x << std::endl;
    return 0;
}

  return num;
}
```

Solution

```
Value of x: 5
```

2- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace FirstNamespace {
    int a = 10;
}

namespace SecondNamespace {
    int b = 20;
}

int main() {
    using namespace FirstNamespace;
    std::cout << "Value of a: " << a << std::endl;

    using namespace SecondNamespace;
    std::cout << "Value of b: " << b << std::endl;

    return 0;
}
```

Solution

```
Value of a: 10
Value of b: 20
```

3- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace VeryLongNamespaceName {
    int value = 42;
}

namespace VLN = VeryLongNamespaceName;

int main() {
    std::cout << "Value: " << VLN::value << std::endl;
    return 0;
}
```

Solution

```
Value: 42
```

4- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace OuterNamespace {
    namespace InnerNamespace {
        int value = 123;
    }
}

int main() {
    std::cout << "Value: " << OuterNamespace::InnerNamespace::value << std::endl;
    return 0;
}
```

Solution

```
Value: 123
```

5- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace MathOperations {
    int add(int a, int b) {
        return a + b;
    }
}

int main() {
    using namespace MathOperations;
    std::cout << "Sum: " << add(3, 4) << std::endl;
    return 0;
}
```

Solution

```
Sum: 7
```

6- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

int globalVar = 10;

namespace MyNamespace {
    int globalVar = 5;
}

int main() {
    std::cout << "Global variable: " << globalVar << std::endl;
    std::cout << "Namespace variable: " << MyNamespace::globalVar << std::endl;
    return 0;
}
```

Solution

```
Global variable: 10
Namespace variable: 5
```

7- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace MyNamespace {
    int x = 42;
}

namespace MyNamespace {
    int y = 21;
}

int main() {
    std::cout << "Values: " << MyNamespace::x << " and " << MyNamespace::y << std::endl;
    return 0;
}
```

Solution

```
Values: 42 and 21
```

8- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace FirstNamespace {
    int x = 5;

    void printX() {
        std::cout << "X from FirstNamespace: " << x << "\n";
    }
}

namespace SecondNamespace {
    int x = 10;

    void printX() {
        std::cout << "X from SecondNamespace: " << x << "\n";
    }
}

int main() {
    FirstNamespace::printX();
    SecondNamespace::printX();

    return 0;
}
```

Solution

```
X from FirstNamespace: 5
X from SecondNamespace: 10
```

9- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace MathOperations {
    int add(int a, int b) {
        return a + b;
    }
}

namespace Display {
    void printResult(int result) {
        std::cout << "Result: " << result << "\n";
    }
}

int main() {
    int sum = MathOperations::add(5, 7);
    Display::printResult(sum);

    return 0;
}
```

Solution

```
Result: 12
```

10- Trace the following program and predict the output.

```cpp
// www.gammal.tech

#include <iostream>

namespace MyNamespace {
    void printMessage() {
        std::cout << "Hello from MyNamespace!\n";
    }
}

int main() {
    MyNamespace::printMessage();

    return 0;
}
```

Solution

```
Hello from MyNamespace!
```
