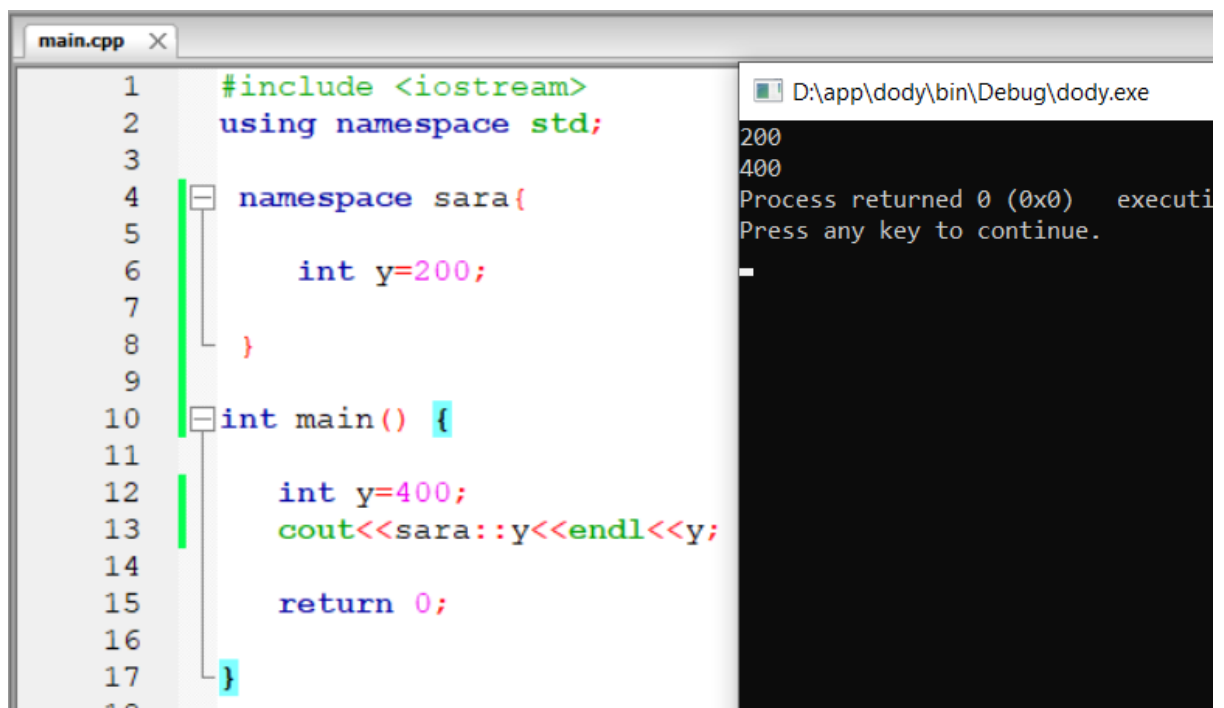# Question: Answer the following questions:

1. What is namespace? Write an example of code?

   Namespace: A namespace is a declarative region that provides a scope to the identifiers (the names of types, functions, variables, etc) inside it.

   Namespaces are used to organize code into logical groups and to prevent name collisions that can occur especially when your code base includes multiple libraries.

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

namespace sara{

    int y=200;

}

int main() {

    int y=400;
    cout<<sara::y<<endl<<y;

    return 0;

}
```

```
D:\app\dody\bin\Debug\dody.exe
200
400
Process returned 0 (0x0)    executi
Press any key to continue.
```

2. What is variable ? write a code in c++ to get the size for some different variables types?

➡ **Variables** are used to store information to be referenced and manipulated in a computer program.

➡ A **variable** in C++ is a **named storage location** in memory that holds a value which can change during program execution.

➡ Code example :

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

int main() {
    // Variable declarations and initializations
    int age = 25;
    float price = 19.99;
    double distance = 12345.6789;
    char grade = 'A';
    bool isStudent = true;
    long population = 8000000;
    short day = 7;
    long long bigNumber = 9876543210;

    // Display values and their sizes
    cout << "Variable      | Value         | Size (bytes)" << endl;
    cout << "-------------------------------------------------" << endl;
    cout << "age           | " << age << "            | " << sizeof(age) << endl;
    cout << "price         | " << price << "          | " << sizeof(price) << endl;
    cout << "distance      | " << distance << "   | " << sizeof(distance) << endl;
    cout << "grade         | " << grade << "            | " << sizeof(grade) << endl;
    cout << "isStudent     | " << isStudent << "            | " << sizeof(isStudent) << endl;
    cout << "population    | " << population << "       | " << sizeof(population) << endl;
    cout << "day           | " << day << "            | " << sizeof(day) << endl;
    cout << "bigNumber     | " << bigNumber << "   | " << sizeof(bigNumber) << endl;

    return 0;
}
```

🖥️ Example Output (on a 64-bit system)

```markdown
Variable        | Value           | Size (bytes)
---------------------------------------------------
age             | 25              | 4
price           | 19.99           | 4
distance        | 12345.7         | 8
grade           | A               | 1
isStudent       | 1               | 1
population      | 8000000         | 8
day             | 7               | 2
bigNumber       | 9876543210      | 8
```

3.What is Variable Initialization and declaration? With an example?

◆ 1. Variable Declaration

When you **declare** a variable, you tell the compiler **what type of data** the variable will hold.

Syntax:

```cpp
data_type variable_name;
```

Example:

```cpp
int age;
float price;
char grade;
```

◆ 2. Variable Initialization

When you **initialize** a variable, you **assign a value** to it (either at declaration time or later).

Syntax:

```cpp
variable_name = value;       // Assign later
// or
data_type variable_name = value;   // Declare + initialize together
```

Example:

```cpp
int age = 20;          // declaration + initialization
float price = 99.99; // declaration + initialization
char grade = 'A';    // declaration + initialization
```

4.What is the overloaded functions ? write example code?

In C++, two functions can have the same name if the number and/or type of arguments passed is different.

These functions having the same name but different arguments are known as overloaded functions.

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

// Function with one int parameter
void display(int a) {
    cout << "Integer value: " << a << endl;
}

// Function with one double parameter
void display(double a) {
    cout << "Double value: " << a << endl;
}

// Function with two parameters
void display(int a, int b) {
    cout << "Sum of integers: " << a + b << endl;
}

int main() {
    display(10);        // Calls display(int)
    display(5.5);       // Calls display(double)
    display(3, 7);      // Calls display(int, int)

    return 0;
}
```

🖥️ Output:

```yaml
Integer value: 10
Double value: 5.5
Sum of integers: 10
```

5. What is the difference between **Local and Global Variable?**

Parameter	Local	Global
Scope	It is declared inside a function.	It is declared outside the function.
Value	If it is not initialized, a garbage value is stored	If it is not initialized zero is stored as default.
Lifetime	It is created when the function starts execution and lost when the functions terminate.	It is created before the program's global execution starts and lost when the program terminates.
Data sharing	Data sharing is not possible as data of the local variable can be accessed by only one function.	Data sharing is possible as multiple functions can access the same global variable.
Modification of variable value	When the value of the local variable is modified in one function, the changes are not visible in another function.	When the value of the global variable is modified in one function changes are visible in the rest of the program.
Accessed by	Local variables can be accessed with the help of statements, inside a function in which they are declared.	You can access global variables by any statement in the program.

6. **Write example code for Local and Global Variable?**

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

// Global variable
int globalVar = 100;

void showValues() {
    // Local variable (only inside this function)
    int localVar = 50;

    cout << "Inside showValues() Function:" << endl;
    cout << "Local Variable = " << localVar << endl;
    cout << "Global Variable = " << globalVar << endl;
}

int main() {
    // Another local variable in main()
    int localVar = 10;

    cout << "Inside main() Function:" << endl;
    cout << "Local Variable = " << localVar << endl;
    cout << "Global Variable = " << globalVar << endl;

    // Call function
    showValues();

    return 0;
```

🖥️ **Example Output**

```java
Inside main() Function:
Local Variable = 10
Global Variable = 100
Inside showValues() Function:
Local Variable = 50
Global Variable = 100
```

7. What are the function types , write an example code?

➡ Predefined Functions
➡ User-Defined Functions

❇ Example: Predefined Function

```cpp
#include <iostream>
#include <cmath>    // For math functions like sqrt, pow, etc.
using namespace std;

int main() {
    double number = 25.0;

    // Predefined functions
    cout << "Square root of " << number << " = " << sqrt(number) << endl;
    cout << "Power (2^3) = " << pow(2, 3) << endl;
    cout << "Absolute value of -10 = " << abs(-10) << endl;

    return 0;
}
```

🖥 Output

```mathematica
Square root of 25 = 5
Power (2^3) = 8
Absolute value of -10 = 10
```

❇️ Example: User-Defined Function

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

// Function declaration (prototype)
int addNumbers(int a, int b);

// Main function
int main() {
    int num1 = 5, num2 = 7;

    // Function call
    int sum = addNumbers(num1, num2);

    cout << "Sum = " << sum << endl;

    return 0;
}

// Function definition
int addNumbers(int a, int b) {
    return a + b;
}
```

🖥️ Output

```ini
Sum = 12
```

8. Create a Calculator using the if Statement?

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

int main() {
    double num1, num2, result;
    char op;

    cout << "Enter first number: ";
    cin >> num1;

    cout << "Enter operator (+, -, *, /): ";
    cin >> op;

    cout << "Enter second number: ";
    cin >> num2;
```

```cpp
// Calculator using if statements
if (op == '+') {
    result = num1 + num2;
    cout << "Result = " << result << endl;
}
else if (op == '-') {
    result = num1 - num2;
    cout << "Result = " << result << endl;
}
else if (op == '*') {
    result = num1 * num2;
    cout << "Result = " << result << endl;
}
else if (op == '/') {
    if (num2 != 0)
        result = num1 / num2;
    else {
        cout << "Error! Division by zero is not allowed." << endl;
        return 0;
    }
    cout << "Result = " << result << endl;
}
else {
    cout << "Invalid operator!" << endl;
}

return 0;
}
```

🖥 Example Output

```sql
Enter first number: 10
Enter operator (+, -, *, /): *
Enter second number: 5
Result = 50
```

Question: Choose the correct answer :

1. What is the correct syntax to output "Hello" in C++?

A) cout << "Hello";
B) Console. WriteLine("Hello");
C) print("Hello");
D) printf("Hello");

2. What is the size of int on most 64-bit systems?

A) 2 bytes
B) 4 bytes
C) 6 bytes
D) 8 bytes

3. Which header file is used for input/output in C++?

A) <stdio.h>
B) <iostream>
C) <conio.h>
D) <string.h>

4. What will be the output of this code?

```cpp
int a = 5, b = 2;
cout << a / b;
```

A) 2.5
B) 2
C) 3
D) Error

5. What type of control structure is an if statement?

A) Loop
B) Decision making
C) Jump
D) Function

6. What is the output of the following code?

```cpp
int i = 1;
while (i <= 3) {
    cout << i << " ";
    i++;
}
```

A) 1 2
B) 1 2 3
C) 1 2 3 4
D) Infinite loop

7. How many times will this loop execute?

```cpp
for (int i = 0; i < 5; i++);
```

A) 4 times
B) 5 times
C) 0 times
D) It does nothing

✔ **Answer:** D

☞ The semicolon ; ends the for loop immediately, so it runs 5 times but does nothing.

8. Which keyword is used to exit a loop immediately?

A) exit

B) stop

C) break

D) return

9. Which statement correctly calls a function named display() with no parameters?

A) call display;

B) display;

C) display();

D) display;()

10. What is the output of this code?

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 2; j++) {
        cout << i << j << " ";
    }
}
```

A) 11 12 21 22 31 32

B) 1 2 3 4 5 6

C) 12 22 32

D) 11 22 33

11. Which statement correctly calls a function named sum with two arguments?

A) sum a, b;

B) sum(a, b);

C) sum a+b;

D) call sum(a, b);

12. Which of the following statements about local variables is true?

A) They are declared outside all functions.
B) They can be used in any function.
C) They are declared inside functions and accessible only there.
D) They have the same value in all functions.