

Assignment

Programming Fundamentals

Submitted by:

Ayesha Sattar

Submitted to:

Mam Sadia Saher

Roll No:

2579

Department:

Computer Science

Q1 — Match the C++ statements with the type of error:

1. cout "Hello"; → **a.** Missing operator (<<)
 2. cin << a; → **b.** Wrong direction of extraction (should be cin >> a;)
 3. int 2num = 5; → **c.** Invalid identifier name (identifier cannot start with digit)
 4. cout << "Hello\World; (missing closing quote in the screenshot) → **d.** Missing quotation mark
 5. if(x>0); cout<<"Yes"; (the screenshot shows an if with an extra semicolon) → **e.** Logical error (semicolon after if)
 6. cout << a + b >> c; → **f.** Invalid operator usage (mixing << and >> incorrectly)
-

Q2 — Output of conditional statements:

- **(True/False):**

1. (14 > 6) → **True**
2. (12 >= 11) → **True**
3. !(a == b) — (if that is the expression shown) then the result depends on a and b; cannot decide without values.
4. (12%3 == 0 || 12%5 == 0) → 12%3 == 0 is true, 12%5 == 0 is false ⇒ overall **True**
5. a = 10, b = 20, c = 15; (a > b && a > c) → 10 > 20 is false ⇒ **False**
6. (10 != 5*2) → 5*2 = 10 so 10 != 10 is **False**

Q3 — Which data type will you select for the following values:

1. “25.75” → **string**
2. 23 → **int**
3. ‘A’ → **char**
4. “Ali” → **string**

Q4 – Trace Output

1.

```
Cout << "Hello\tWorld\n";
Cout << "C++\\Program\\ning";
Cout << "C++\\bProgram\\bning";
```

Output:

```
Hello  World
C++\Program
ning
C+Prograning
```

2.

```
Int a = 10, b = 4;
Cout << a % b;
```

Output:

```
2
```

3.

```
Int x = 3;
X = x + 4;
Cout << x + 2;
```

```
X = 7
X + 2 = 9
```

Output:

```
9
```

4.

```
Int a = 2.3;  
Cout << a;
```

Output:

2

5.

```
Int x=3, y=4, z=5;  
If(x>y)  
    Z+=x;  
else if(y>z)  
    Z=z-y;  
else  
    Z=z*x;  
Cout << z;
```

x>y? → 3>4 → false
y>z? → 4>5 → false
else: z = 5 * 3 = 15

Output:

15

6.

```
Cout << "Enter age: ";  
Cin >> age;  
Cout << "Your age is " << age;
```

Output:

Enter age: Your age is 18

7.

```
Int a = 5;  
If(a > 5)  
    cout << "A";  
else  
    cout << "B";
```

$5 > 5 \rightarrow \text{false}$

Output:

B

8.

```
Int x = 2, y = 3, z = 4;a  
cout << x + y * z / x;
```

Output:

8

9.

```
Int a = 5, b = 10;  
Cout << (a > b ? a : b);
```

Output:

10

10.

```
int x = 2, y = 3;  
cout << (x+1*y+1) + (y-1);
```

Output:

8

11.

```
float 1rate = 10.5;  
cout << 1rate;
```

Output:

10.5

12.

```
If(a = 5)  
Cout << "Five";
```

Output:

Five

Error: (a is not initialized)

13.

```
int a;  
cout << a;
```

A is garbage value

Exact output cannot be predicted.

Output:

Garbage value

14.

```
cout << "She said \"Hello\"";
```

Output:

She said “Hello”

15.

```
int n = 5;
```

Output:

(no output)

16.

```
int num = "5";
```

Output:

(no output)

18.

```
cout << "Area = ";
```

```
cout << 3.14 * r * r;
```

r is not declared → **compile error**

Output:

Error / no output

19.

```
if(x > 0) {  
    cout << "Positive";  
}
```

- x (not given)

Output:

Positive

20.

```
cin >> number;  
if(number == 10)  
    cout << "Equal";
```

Output:

Equal // only if input = 10

21.

```
if(x < 5 && x >= 2)  
    cout << "Range";
```

Output:

Range

22.

```
int result = 10 / 2 *;
```

syntax error (missing operand).

Output:

Error

23.

```
cout << 1 + 4 >> 7;
```

Output:

0

24.

```
cout << "Path: C:\newfolder";
```

Output:

Path: C:
ewfolder

Q5. Short Questions — Answers

1. Explain the difference between cin and cout in C++.

Cin is used for input (takes data from the keyboard).

Cout is used for output (displays data on the screen).

2. What happens if a user inputs a value of the wrong data type using cin?

The input operation fails, cin goes into a fail state, and the variable does not receive the value. The program may skip further input operations.

3. Differentiate between float and double.

Float is a single-precision data type (4 bytes).

Double is a double-precision data type (8 bytes) with more accuracy.

4. What happens if you use an uninitialized variable in an expression?

It contains garbage value, so the result will be unpredictable.

5. What is an escape sequence in C++ and why is it needed?

An escape sequence is a special character combination starting with \ used to represent actions like newline, tab, or printing special characters.

Example: \n, \t, \".

6. Why can't we simply press Enter inside a string to create a new line instead of using “\n”?

Strings cannot contain a physical line break. \n is needed to instruct the compiler to move the output to the next line.

7. Explain the purpose of the modulus (%) operator.

It gives the remainder of a division.

Example: $10 \% 3 = 1$.

8. Can else be used without a preceding if? Explain.

No.

else must always be paired with a previous if statement.

9. What happens if you end an if condition with a semicolon?

The semicolon terminates the if statement, so the next block executes unconditionally.

This is a logical error.

10. Explain the difference between == and = operators.

== is a comparison operator (checks equality).

= is an assignment operator (stores a value in a variable).

11. Can you combine multiple relational conditions in one expression? Give an example.

Yes.

Example: $(a > b \&\& b > c)$.

12. What happens when operators have the same precedence?

The expression is evaluated according to associativity (left-to-right or right-to-left).

13. Write the general syntax of the ternary operator and explain its components.

Syntax:

Condition ? expression1 : expression2;

If condition is true \rightarrow expression1 executes

If false \rightarrow expression2 executes

14. What are the naming rules for identifiers in C++?

Must start with a letter or underscore

Cannot start with a digit

Cannot contain spaces

Cannot use reserved keywords

Can contain letters, digits, and underscores

Q6 — Program

Problem:

Check whether a year is between 2000–2025.

If 2000–2015 → print sum of 2000 & 2015

If 2016–2025 → print sum of 2016 & 2025

else → “Year is not in the range”

C++ Program:

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

int main() {

    int year;
    cout << "Enter year: ";
    cin >> year;

    if (year >= 2000 && year <= 2015) {
        cout << "Sum = " << 2000 + 2015;
    }
    else if (year >= 2016 && year <= 2025) {
        cout << "Sum = " << 2016 + 2025;
    }
    else {
        cout << "Year is not in the range.";
    }
}
```

```
    return 0;  
}
```
