



ECE 201 Quiz - Extra Examples

Based on Lecture 1, Lecture 2

Section 1: Code Tracing – Predict the Output

Q1.

```
#include <stdio.h>
int main(void) {
    int x = 5;
    int y = 10;
    int sum = x + y;
    printf("%d", sum);
    return 0;
}
```

Choices:

- A) 15
- B) x + y
- C) Error: undeclared variable
- D) 510

Answer: A) 15

Q2.

```
#include <stdio.h>
int main(void) {
    int a = 3, b = 4;
    printf("Sum is %d\n", a + b);
    return 0;
}
```

Choices:

- A) Sum is 7
- B) Sum is a + b
- C) Error: missing semicolon
- D) Sum is 34

Answer: A) Sum is 7

Q3.

```
#include <stdio.h>
int main(void) {
    int a = 2;
    int b = 3;
    int result = a * b + 1;
    printf("%d", result);
    return 0;
}
```

Choices:

- A) 5
- B) 7
- C) 6
- D) Error

Answer: B) 7

Q4.

```
#include <stdio.h>
int main(void) {
    int x = 8;
    printf("Value: %d\n", x);
    return 0;
}
```

Choices:

- A) Value: x
- B) Value: %d
- C) Value: 8
- D) Error

Answer: C) Value: 8

Q5.

```
#include <stdio.h>
int main(void) {
    int a = 4;
    int b = 2;
    int c = a / b;
    printf("%d", c);
    return 0;
}
```

Choices:

- A) 2
- B) 0
- C) 1
- D) Error

Answer: A) 2

Q6.

```
#include <stdio.h>
int main(void) {
    int a = 10;
    int b = 20;
    int c = b - a;
    printf("%d", c);
    return 0;
}
```

Choices:

- A) 30
- B) 10
- C) 20
- D) 0

Answer: B) 10

Q7.

```
#include <stdio.h>
int main(void) {
    int x = 7;
    int y = 3;
    int z = x % y;
    printf("%d", z);
    return 0;
}
```

Choices:

- A) 2
- B) 1
- C) 3
- D) 0

Answer: B) 1

Q8.

```
#include <stdio.h>
int main(void) {
    int a = 5;
    a = a + 2;
    printf("%d", a);
    return 0;
}
```

Choices:

- A) 2
- B) 7
- C) 5
- D) 10

Answer: B) 7

Q9.

```
#include <stdio.h>
int main(void) {
    int x = 4;
    int y = 2;
    int result = x / y;
    printf("%d", result);
    return 0;
}
```

Choices:

- A) 0
- B) 4
- C) 1
- D) 2

Answer: D) 2

Q10.

```
#include <stdio.h>
int main(void) {
    int a = 1;
    int b = 2;
    int c = 3;
    int result = a + b * c;
}
```

```
    printf("%d", result);  
    return 0;  
}
```

Choices:

- A) 7
- B) 5
- C) 9
- D) 6

Answer: A) 7

Q11.

```
#include <stdio.h>  
int main(void) {  
    float x = 3.5;  
    float y = 2.0;  
    float result = x + y;  
    printf("%.1f", result);  
    return 0;  
}
```

Choices:

- A) 6.0
- B) 7.0
- C) 3.0
- D) 5.5

Answer: D) 5.5

Q12.

```
#include <stdio.h>  
int main(void) {  
    char ch = 'A';  
    printf("%c", ch);  
    return 0;  
}
```

Choices:

- A) A
- B) %c
- C) 65
- D) Error

Answer: A) A

Q13.

```
#include <stdio.h>
int main(void) {
    double pi = 3.14159;
    printf("%.2f", pi);
    return 0;
}
```

Choices:

- A) 3.1
- B) 3.14
- C) 3.142
- D) 3.14159

Answer: B) 3.14

Q14.

```
#include <stdio.h>
int main(void) {
    float a = 5.0;
    int b = 2;
    float result = a / b;
    printf("%.2f", result);
    return 0;
}
```

Choices:

- A) 2.50
- B) 2
- C) 2.0
- D) 2.25

Answer: A) 2.50

Q15.

```
#include <stdio.h>
int main(void) {
    float a = 2.5;
    float b = 4.0;
    float result = a * b;
    printf("%.2f", result);
    return 0;
}
```

Choices:

- A) 6.5
- B) 10.0
- C) 8.0
- D) 6.0

Answer: C) 10.00

Q16.

```
#include <stdio.h>
int main(void) {
    double x = 1.234;
    double y = 2.0;
    double z = x + y;
    printf("%.3f", z);
    return 0;
}
```

Choices:

- A) 3.234
- B) 3.23
- C) 3.2
- D) 3.000

Answer: A) 3.234

Q17.

```
#include <stdio.h>
int main(void) {
    char first = 'C';
    char second = 'D';
    printf("%c %c", first, second);
    return 0;
}
```

Choices:

- A) CD
- B) C D
- C) %c %c
- D) Error

Answer: B) C D

Q18.

```
#include <stdio.h>
int main(void) {
    float temp = 36.6;
    printf("Temperature: %.1f", temp);
    return 0;
}
```

Choices:

- A) Temperature: 36.60
- B) Temperature: 36.6
- C) Temperature: 36
- D) Error

Answer: B) Temperature: 36.6

Q19.

```
#include <stdio.h>
int main(void) {
    double a = 5.0;
    double b = 2.0;
    double result = a / b;
    printf("%.2f", result);
    return 0;
}
```

Choices:

- A) 2.50
- B) 2.5
- C) 2
- D) 2.00

Answer: D) 2.50

Section 2: Error Identification (Continued)

T20.

```
#include <stdio.h>
int main(void) {
    int number;
    number = '100';
    printf("%d", number);
    return 0;
}
```

Choices:

- A) Prints 100
- B) Error: character assigned to int
- C) Prints ASCII value
- D) Error: missing semicolon

Answer: B) Error: character assigned to int

Fix:

```
number = 100;
```

T21.

```
#include <stdio.h>
int main(void) {
    float rate;
    rate = '3.5';
    printf("%.2f", rate);
    return 0;
}
```

Choices:

- A) Prints 3.5
- B) Error: type mismatch
- C) Prints ASCII value
- D) Prints 0.00

Answer: B) Error: type mismatch

Fix:

```
rate = 3.5;
```

T22.

```
#include <stdio.h>
int main(void) {
    int total;
    total = 50;
    printf("%f", total);
    return 0;
}
```

Choices:

- A) Prints 50.00
- B) Error: format specifier mismatch
- C) Prints garbage value
- D) Prints 0.00

Answer: B) Error: format specifier mismatch**Fix:**

```
printf("%d", total);
```

T23.

```
#include <stdio.h>
int main(void) {
    int sum;
    sum = 25 + ;
    printf("%d", sum);
    return 0;
}
```

Choices:

- A) Prints 25
- B) Error: incomplete expression
- C) Prints 0
- D) Error: undeclared variable

Answer: B) Error: incomplete expression**Fix:**

```
sum = 25 + 5;
```

T24.

```
#include <stdio.h>
int main(void) {
    int x;
    x = 10
    printf("%d", x);
    return 0;
}
```

Choices:

- A) Prints 10
- B) Error: missing semicolon
- C) Prints 0
- D) Error: undeclared variable

Answer: B) Error: missing semicolon**Fix:**

```
x = 10;
```

T25.

```
#include <stdio.h>
int main(void) {
    int sum;
    int total;
    sum = 50;
    total = sum + unknown;
    printf("%d", total);
    return 0;
}
```

Choices:

- A) Prints 50
- B) Error: undeclared variable
- C) Prints garbage value
- D) Prints 0

Answer: B) Error: undeclared variable**Fix:**

```
int unknown = 10;
```

T26.

```
#include <stdio.h>
int main(void) {
    int 1stNumber = 5;
    printf("%d", 1stNumber);
    return 0;
}
```

Choices:

- A) Prints 5
- B) Error: invalid identifier
- C) Error: missing semicolon
- D) Prints 0

Answer: B) Error: invalid identifier**Fix:**

```
int firstNumber = 5;
```

T27.

```
#include <stdio.h>
int main(void) {
    char letter;
    float pi;
    letter = 'A';
    pi = "3.14";
    printf("Letter: %c, Pi: %f", letter, pi);
    return 0;
}
```

Choices:

- A) Prints: Letter: A, Pi: 3.14
- B) Error: type mismatch
- C) Error: invalid format specifier
- D) Prints: Letter: A, Pi: 0.00

Answer: B) Error: type mismatch**Fix:**

```
pi = 3.14;
```

T28.

```
#include <stdio.h>
int main(void) {
    int x = 10
    printf("%d", x);
    return 0;
}
```

Choices:

- A) Prints 10
- B) Error: missing semicolon
- C) Error: undeclared variable
- D) Prints 0

Answer: B) Error: missing semicolon**Fix:**

```
int x = 10;
```

T29.

```
#include <stdio.h>
int main(void) {
    int value;
    printf("%d", value);
    return 0;
}
```

Choices:

- A) Prints 0
- B) Error: uninitialized variable
- C) Prints garbage value
- D) Error: undeclared variable

Answer: B) Error: uninitialized variable**Fix:**

```
int value = 0;
```

T30.

```
#include <stdio.h>
int main(void) {
    int a = 5;
    int b = 10;
```

```
int sum = a + b
printf("%d", sum);
return 0;
}
```

Choices:

- A) Prints 15
- B) Error: missing semicolon
- C) Error: undeclared variable
- D) Prints garbage value

Answer: B) Error: missing semicolon

Fix:

```
int sum = a + b;
```

T31.

```
#include <stdio.h>
int main(void) {
    int total = 100;
    printf("%f", total);
    return 0;
}
```

Choices:

- A) Prints 100.00
- B) Error: format specifier mismatch
- C) Prints garbage value
- D) Prints 0.00

Answer: B) Error: format specifier mismatch

Fix:

```
printf("%d", total);
```

T32.

```
#include <stdio.h>
int main(void) {
    int 2ndValue = 20;
    printf("%d", 2ndValue);
    return 0;
}
```

Choices:

- A) Prints 20
- B) Error: invalid identifier
- C) Error: missing semicolon
- D) Prints 0

Answer: B) Error: invalid identifier

Fix:

```
int secondValue = 20;
```

T33.

```
#include <stdio.h>
int main(void) {
    float pi = "3.14";
    printf("%.2f", pi);
    return 0;
}
```

Choices:

- A) Prints 3.14
- B) Error: type mismatch
- C) Prints garbage value
- D) Prints 0.00

Answer: B) Error: type mismatch

Fix:

```
float pi = 3.14;
```

T34.

```
#include <stdio.h>
int main(void) {
    int x = 5;
    int y = 10;
    int result = x + y;
    printf("Result is: %d %d", result);
    return 0;
}
```

Choices:

- A) Prints: Result is: 15

- B) Error: missing argument for format specifier
- C) Prints: Result is: 15 0
- D) Error: undeclared variable

Answer: B) Error: missing argument for format specifier

Fix:

```
printf("Result is: %d", result);
```

T35.

```
#include <stdio.h>
int main(void) {
    int sum;
    sum = 25 + ;
    printf("%d", sum);
    return 0;
}
```

Choices:

- A) Prints 25
- B) Error: incomplete expression
- C) Prints 0
- D) Error: undeclared variable

Answer: B) Error: incomplete expression

Fix:

```
sum = 25 + 5;
```

T36.

```
#include <stdio.h>
int main(void) {
    char grade;
    scanf("%c", grade);
    printf("Grade: %c\n", grade);
    return 0;
}
```

Choices:

- A) Correct usage
- B) Error: missing &
- C) Error: wrong format specifier

- D) Error: undeclared variable

Answer: B) Error: missing &

Fix:

```
scanf ("%c", &grade);
```



Section 3: Comprehensive Output Tracing

```
1 /* Line 1: Program demonstrating Lecture 1 & 2 concepts */
2 #include <stdio.h> // Line 2: Preprocessor directive
3
4 int main(void) { // Line 3: Main function
5
6 // Line 4 8 : Variable declarations
7 int number1 = 10; // Line 4
8 float pi = 3.14; // Line 5
9 double gravity = 9.80665; // Line 6
10 char grade = 'A'; // Line 7
11 const int MAX = 100; // Line 8: Constant declaration
12
13 // Line 9 11 : Output using format specifiers
14 printf("Integer: %d\n", number1); // Line 9
15 printf("Float: %.2f\n", pi); // Line 10
16 printf("Double: %.5f\n", gravity); // Line 11
17
18 // Line 12 13 : Character and constant output
19 printf("Grade: %c\n", grade); // Line 12
20 printf("Max value: %d\n", MAX); // Line 13
21
22 // Line 14: Escape sequences
23 printf("Line break\\n and tab\\t test:\\n\\tDone\\n"); // Line 14
24
25 // Line 15 17 : Input and output
26 int input; // Line 15
27 printf("Enter a number: "); // Line 16
28 scanf("%d", &input); // Line 17
29 printf("You entered: %d\n", input); // Line 18
30
31 return 0; // Line 19
32 }
```

Output Explanation by Line

- Line 9: Integer: 10
- Line 10: Float: 3.14
- Line 11: Double: 9.80665
- Line 12: Grade: A
- Line 13: Max value: 100

- **Line 14:** Line break and tab test: followed by a tabbed line: Done
- **Line 16:** Prompts user: Enter a number:
- **Line 17:** Accepts user input (e.g., 42)
- **Line 18:** Echoes input: You entered: 42

```
1  /* Program: Basic Operations and Output Formatting */
2  #include <stdio.h>                // Line 2: Preprocessor directive
3
4  int main(void) {                  // Line 3: Main function
5
6      // Line 4 8 : Variable declarations
7      int a = 12;                   // Line 4
8      int b = 8;                    // Line 5
9      float average;                // Line 6
10     double precise = 3.1415926536; // Line 7
11     char symbol = '#';            // Line 8
12
13     // Line 9 11 : Arithmetic operations
14     int sum = a + b;               // Line 9
15     int product = a * b;          // Line 10
16     average = (a + b) / 2.0;      // Line 11
17
18     // Line 12 14 : Output formatting
19     printf("Sum: %d\n", sum);      // Line 12
20     printf("Product: %d\n", product); // Line 13
21     printf("Average: %.2f\n", average); // Line 14
22
23     // Line 15 16 : Character and double output
24     printf("Symbol: %c\n", symbol); // Line 15
25     printf("Precise value: %.10f\n", precise); // Line 16
26
27     // Line 17 19 : Escape sequences
28     printf("Tab\tSpace and newline\nTest:\n\tOK\n"); // Line 17
29
30     // Line 20 23 : Input and echo
31     int input1, input2;
32     printf("Enter two integers: "); // Line 21
33     scanf("%d %d", &input1, &input2); // Line 22
34     printf("You entered: %d and %d\n", input1, input2); // Line 23
35
36     return 0;                      // Line 24
37 }
```

Output Explanation by Line

- **Line 12:** Sum: 20
- **Line 13:** Product: 96
- **Line 14:** Average: 10.00
- **Line 15:** Symbol: #
- **Line 16:** Precise value: 3.1415926536
- **Line 17:** Taband newline: followed by tabbed line: OK
- **Line 21:** Prompts user: Enter two integers:
- **Line 22:** Accepts user input (e.g., 5 7)
- **Line 23:** Echoes input: You entered: 5 and 7

Section 4: Output Tracing and Error Identification

```
1 /* Program: Mixed Data Types and Output */
2 #include <stdio.h> // Line 2
3
4 int main(void) { // Line 3
5
6     int x = 10; // Line 4
7     float y = 5.5; // Line 5
8     double z = 2.71828; // Line 6
9     char grade = 'B'; // Line 7
10    const int MAX = 100; // Line 8
11
12    int result = x + y; // Line 9
13    printf("Result: %d\n", result); // Line 10
14
15    printf("Float value: %f\n", y); // Line 11
16    printf("Double value: %.3f\n", z); // Line 12
17    printf("Grade: %c\n", grade); // Line 13
18
19    int 1stPlace = 1; // Line 14 ERROR
20    printf("First place: %d\n", 1stPlace); // Line 15
21
22    int total;
23    total = "50"; // Line 17 ERROR
24    printf("Total: %d\n", total); // Line 18
25
26    printf("Max constant: %d\n", MAX); // Line 19
27
```

```
28     printf("Tab\\tTest\\n\\tDone\\n"); // Line 20
29
30     return 0; // Line 21
31 } // Line 22
```

Instructions

For each line that produces output, write the expected result. For each line that contains an error, describe the error and suggest a fix.

Expected Output and Error Analysis

- **Line 10:** ‘Result: 15’ *Note: Implicit conversion from float to int truncates decimal.*
- **Line 11:** ‘Float value: 5.500000’
- **Line 12:** ‘Double value: 2.718’
- **Line 13:** ‘Grade: B’
- **Line 14:** ****Error: Invalid identifier**** *Fix: Variable names cannot start with digits. Use:*

```
int firstPlace = 1;
```

- **Line 15:** ****Error: Undeclared variable**** *Fix: Update to match corrected identifier:*

```
printf("First place: %d\\n", firstPlace);
```

- **Line 17:** ****Error: Type mismatch**** *Fix: Assign integer literal, not string:*

```
total = 50;
```

- **Line 18:** ‘Total: 50’ (after fix)
- **Line 19:** ‘Max constant: 100’
- **Line 20:** Tab Test Done
- **Line 21:** Program ends successfully.

Section 5: Inputs, Outputs, and Common Errors

```
1  /* Program: Student Grade Calculator with Errors and Fixes */
2  #include <stdio.h>                // Line 2: Preprocessor directive
3
4  int main(void) {                  // Line 3: Main function
5
6      // Line 4 8 : Variable declarations
7      int studentID;                // Line 4
8      char initialName;            // Line 5
9      float quiz1, quiz2;          // Line 6
10     double average;               // Line 7
11     const int MAX_SCORE = 100;    // Line 8
12
13     // Line 9 12 : Input section
14     printf("Enter student ID: "); // Line 9
15     scanf("%d", studentID);      // Line 10 Error: missing
16     &
17     printf("Enter student initial name: "); // Line 11
18     scanf("%c", initialName);    // Line 12
19
20     // Line 13 16 : Quiz scores
21     printf("Enter Quiz 1 score: "); // Line 13
22     scanf("%f", &quiz1);         // Line 14
23
24     printf("Enter Quiz 2 score: "); // Line 15
25     scanf("%d", &quiz2);         // Line 16 Error: wrong
26     format specifier
27
28     // Line 17 18 : Calculate average
29     average = (quiz1 + quiz2) / 2.0; // Line 17
30     printf("Average score: %.2f\n", average); // Line 18
31
32     // Line 19 21 : Character grade
33     char grade;
34     printf("Enter grade ( A F ): "); // Line 20
35     scanf("%c", &grade);         // Line 21 Error: may
36     capture newline
37
38     // Line 22 24 : Output summary
39     printf("\nStudent Summary:\n"); // Line 22
40     printf("ID: %d\n", studentID); // Line 23
41     printf("Initial Name: %c\n", initialName); // Line 24
42
43     // Line 25 27 : Scores and grade
44     printf("Quiz 1: %.2f\n", quiz1); // Line 25
```

```
43     printf("Quiz 2: %.2f\n", quiz2);           // Line 26
44     printf("Grade: %c\n", grade);           // Line 27
45
46     // Line 28 30 : Escape sequence demo
47     printf("Max score is %d\n", MAX_SCORE); // Line 28
48     printf("Tab\\tTest\n\\tDone\n");      // Line 29
49
50     // Line 31 33 : Invalid identifier example
51     int 1stRank = 1;                       // Line 31 Error:
52         identifier starts with digit
53     printf("Rank: %d\n", 1stRank);         // Line 32 Error:
54         undeclared variable
55
56     return 0;                             // Line 33
57 }                                           // Line 34
```

Error Summary and Fixes

- **Line 10:** Missing & before variable **Fix:** `scanf("%d", &studentID);`
- **Line 16:** Wrong format specifier for float **Fix:** `scanf("%f", &quiz2);`
- **Line 21:** May capture newline from previous input **Fix:** `scanf(" %c", &grade);`
- **Line 31:** Invalid identifier (starts with digit) **Fix:** `int firstRank = 1;`
- **Line 32:** Undeclared variable (due to invalid name) **Fix:** `printf("Rank: %d\n", firstRank);`

Expected Output (After Fixes)

```
Enter student ID: 123
Enter student initial name: S
Enter Quiz 1 score: 85.5
Enter Quiz 2 score: 90.0
Enter grade (A{F}): A
```

Student Summary:

```
ID: 123
Initial Name: S
Quiz 1: 85.50
Quiz 2: 90.00
Grade: A
Max score is 100\n
Done
Rank: 1
```

End of Quiz — Great work!

