

# Mock Exam 2: Variables, Control Flow, and Functions

---

**Time Limit: 1 hour**

**Total Marks: 100**

---

## Section A: Multiple Choice Questions (30 marks)

*Choose the best answer for each question. 2 marks each.*

**Question 1:** What is the correct way to declare and initialize an integer variable in C? a) `int x = 5;` b) `integer x = 5;` c) `int x := 5;` d) `var x = 5;`

**Question 2:** What will this code print?

```
int x = 10;
if (x > 5 && x < 15) {
    printf("Medium");
} else {
    printf("Other");
}
```

a) Medium b) Other c) Nothing d) Error

**Question 3:** Which loop is best for counting from 1 to 10? a) `while` loop only b) `for` loop only  
c) Both work equally well d) Neither works for counting

**Question 4:** What does this function do?

```
int mystery(int a, int b) {
    if (a > b) return a;
    else return b;
}
```

a) Returns the sum of a and b b) Returns the larger of a and b c) Returns the smaller of a and b d) Returns the difference between a and b

**Question 5:** What is the output of this code?

```
int count = 0;
while (count < 3) {
    printf("%d ", count);
    count++;
}
```

a) 0 1 2 b) 1 2 3 c) 0 1 2 3 d) Infinite loop

**Question 6:** Which is NOT a valid data type in C? a) `int` b) `double` c) `string` d) `char`

**Question 7:** What makes a function a "void" function? a) It has no parameters b) It returns no value c) It prints nothing d) It cannot be called

**Question 8:** What will `x` be after this code runs?

```
int x = 5;
x += 3;
x *= 2;
```

a) 10 b) 13 c) 16 d) 11

**Question 9:** Which operator is used for "not equal to" in C? a) `<>` b) `!=` c) `=/=` d) `!==`

**Question 10:** What's the correct way to define a constant? a) `const int MAX = 100;` b) `#define MAX 100` c) `final int MAX = 100;` d) Both a and b are correct

**Question 11:** In pass-by-value, what happens to the original variable? a) It gets modified b) It remains unchanged c) It gets deleted d) It becomes a copy

**Question 12:** What does the `%` operator do? a) Percentage calculation b) Modulo (remainder) operation c) Division d) Multiplication

**Question 13:** Which is the correct structure for an if-else statement? a) `if (condition) { } else { }` b) `if condition { } else { }` c) `if (condition) then { } else { }` d) `if (condition): { } else: { }`

**Question 14:** What will this print?

```
for (int i = 5; i > 2; i--) {
    printf("%d ", i);
}
```

a) 5 4 3 b) 5 4 3 2 c) 3 4 5 d) Error

**Question 15:** Which function prototype is correct for a function that takes two integers and returns nothing? a) `void calculate(int, int);` b) `int calculate(void, void);` c) `calculate(int a, int b);` d) `void calculate();`

---

## Section B: Code Completion (25 marks)

*Complete the missing parts of the code. 5 marks each.*

**Question 16:** Complete this function to return the absolute value of a number:

```
int absolute_value(int num) {  
    if (_____) {  
        return _____;  
    } else {  
        return _____;  
    }  
}
```

**Question 17:** Fill in the blanks to create a countdown loop from 10 to 1:

```
for (int i = _____; _____ >= 1; _____) {  
    printf("%d ", i);  
}
```

**Question 18:** Complete this function to check if a number is even:

```
int is_even(int number) {  
    if (number _____ 2 == 0) {  
        return _____;  
    } else {  
        return _____;  
    }  
}
```

**Question 19:** Fill in the grade calculation logic:

```
char get_grade(int score) {  
    if (score >= _____) {  
        return 'A';  
    } else if (score >= _____) {  
        return 'B';  
    } else if (score >= _____) {  
        return 'C';  
    } else {  
        return '_____';  
    }  
}
```

**Question 20:** Complete this function to calculate the factorial of a number:

```
int factorial(int n) {  
    int result = _____;  
    for (int i = 1; i <= _____; i++) {  
        result _____ i;  
    }  
}
```

```
    }  
    return result;  
}
```

---

## Section C: Write Complete Functions (45 marks)

Write complete, working functions for each problem.

**Question 21: Temperature Converter (15 marks)** Write a function called `celsius_to_fahrenheit` that:

- Takes a double parameter representing temperature in Celsius
- Returns the equivalent temperature in Fahrenheit
- Formula:  $F = (C \times 9/5) + 32$

Also write a `main` function that:

- Prompts the user for a Celsius temperature
- Calls your function to convert it
- Prints the result with 1 decimal place

**Question 22: Number Analyzer (15 marks)** Write a function called `analyze_number` that:

- Takes an integer parameter
- Prints whether the number is:
  - Positive, negative, or zero
  - Even or odd (only if the number is not zero)
  - Single digit (0-9) or multi-digit

Example output for input 15:

```
15 is positive  
15 is odd  
15 is multi-digit
```

**Question 23: Simple Calculator (15 marks)** Write a complete program that:

- Prompts the user for two numbers and an operation (+, -, \*, /)
- Uses a function called `calculate` that takes two doubles and a char for the operation
- Returns the result of the calculation
- Handles division by zero by returning -1
- In main, prints the result or "Error: Division by zero" if result is -1

Example interaction:

```
Enter first number: 10  
Enter operation (+, -, *, /): /
```

```
Enter second number: 3  
Result: 3.33
```

---

## Answer Template

*Use this space to write your code solutions. You may use additional paper if needed.*

### Section C Solutions:

#### Question 21:

```
// Write your celsius_to_fahrenheit function and main function here
```

#### Question 22:

```
// Write your analyze_number function here
```

#### Question 23:

```
// Write your complete calculator program here
```

---

## Marking Rubric

- **MCQ (30 marks):** 2 marks per correct answer
- **Code Completion (25 marks):** 5 marks per question - correct logic and syntax
- **Programming (45 marks):**
  - Correct function signature: 3 marks each
  - Correct logic implementation: 8 marks each
  - Proper input/output handling: 4 marks each

**Good luck! Remember to:**

- Check your syntax carefully
- Test your logic with sample values
- Handle edge cases appropriately
- Write clean, readable code