Assignment 1: C Programming Basics

Smriti Shahi

Westcliff University

C Programs

Nirajan Thakuri

February 7, 2026

Assignment 1: C Programming Basics

1.

main.c — Share — Run — Output

```c
#include <stdio.h>

int main() {
    printf("This is the starting point of the program.\n");
    printf("The main() function is where execution begins.\n");
    return 0;
}
```

```
This is the starting point of the program.
The main() function is where execution begins.


=== Code Execution Successful ===
```

2.

main.c — Share — Run — Output

```c
#include <stdio.h>

int main() {
    int declared_var; // Declaration - holds a garbage value
    int initialized_var = 10; // Initialization - holds the value 10

    printf("Declared variable (garbage value): %d\n", declared_var);
    printf("Initialized variable: %d\n", initialized_var);

    return 0;
}
```

```
Declared variable (garbage value): 1195739856
Initialized variable: 10


=== Code Execution Successful ===
```

3.

main.c — Share — Run — Output

```c
#include <stdio.h>

int main() {
    char name[50];

    printf("Please enter your name: ");
    scanf("%49s", name);

    printf("\nHello %s, welcome to the C programming world!\n", name);

    return 0;
}
```

```
Please enter your name: smriti

Hello smriti, welcome to the C programming world!


=== Code Execution Successful ===
```

4.

main.c

```c
#include <stdio.h>

int main() {
    int integer_var = 42;
    float float_var = 3.14;
    double double_var = 2.71828;
    char char_var = 'A';

    printf("Integer: %d\n", integer_var);
    printf("Float: %.2f\n", float_var);
    printf("Double: %.5f\n", double_var);
    printf("Character: %c\n", char_var);

    return 0;
}
```

Output

```
Integer: 42
Float: 3.14
Double: 2.71828
Character: A


=== Code Execution Successful ===
```

5.

main.c

```c
#include <stdio.h>

int main() {
    // Implicit Conversion
    int num_int = 10;
    double num_double = num_int; // int is implicitly converted to
        double
    printf("Implicit conversion: %d becomes %.1f\n", num_int,
        num_double);

    // Explicit Conversion
    double pi = 3.14159;
    int truncated_pi = (int)pi; // double is explicitly cast to int
    printf("Explicit conversion: %.5f becomes %d\n", pi, truncated_pi
        );

    return 0;
}
```

Output

```
Implicit conversion: 10 becomes 10.0
Explicit conversion: 3.14159 becomes 3


=== Code Execution Successful ===
```

6.

main.c

```c
#include <stdio.h>

int main() {
    float length, width, area;

    printf("Enter the length of the rectangle: ");
    scanf("%f", &length);

    printf("Enter the width of the rectangle: ");
    scanf("%f", &width);

    area = length * width;

    printf("The area of the rectangle is: %.2f\n", area);

    return 0;
}
```

Output

```
Enter the length of the rectangle: 10
Enter the width of the rectangle: 11
The area of the rectangle is: 110.00


=== Code Execution Successful ===
```

7.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int age;
    float salary;

    printf("Enter your age and salary, separated by a space: ");
    scanf("%d %f", &age, &salary);

    printf("You entered: Age = %d, Salary = %.2f\n", age, salary);

    return 0;
}
```

Output

```
Enter your age and salary, separated by a space: 20


20,000
You entered: Age = 20, Salary = 20.00


=== Code Execution Successful ===
```

8.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    float celsius, fahrenheit;

    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);

    fahrenheit = (celsius * 9.0 / 5.0) + 32;

    printf("%.2f Celsius is equal to %.2f Fahrenheit.\n", celsius,
        fahrenheit);

    return 0;
}
```

Output

```
Enter temperature in Celsius: 10
10.00 Celsius is equal to 50.00 Fahrenheit.


=== Code Execution Successful ===
```

9.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int total_days, weeks, remaining_days;

    printf("Enter the total number of days: ");
    scanf("%d", &total_days);

    weeks = total_days / 7;
    remaining_days = total_days % 7;

    printf("%d days = %d week(s) and %d day(s).\n", total_days, weeks,
        remaining_days);

    return 0;
}
```

Output

```
Enter the total number of days: 11
11 days = 1 week(s) and 4 day(s).


=== Code Execution Successful ===
```

10.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int a = 10, b = 20;
    int temp;

    printf("Before swapping: a = %d, b = %d\n", a, b);

    temp = a;
    a = b;
    b = temp;

    printf("After swapping:  a = %d, b = %d\n", a, b);

    return 0;
}
```

Output

```
Before swapping: a = 10, b = 20
After swapping:  a = 20, b = 10


=== Code Execution Successful ===
```

11.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int my_var = 5;
    int result;

    result = ((my_var + 1) * 3) - 10;

    printf("Original variable: %d\n", my_var);
    printf("Result of the expression ((var + 1) * 3) - 10: %d\n",
        result);
    printf("Calculation: ((5 + 1) * 3) - 10 = (6 * 3) - 10 = 18 - 10 =
        8\n");

    return 0;
}
```

Output

```
Original variable: 5
Result of the expression ((var + 1) * 3) - 10: 8
Calculation: ((5 + 1) * 3) - 10 = (6 * 3) - 10 = 18 - 10 = 8


=== Code Execution Successful ===
```

12.

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int a = 15, b = 10, c = 5;

    printf("a = %d, b = %d, c = %d\n", a, b, c);

    if (a > b && c != 0) {
        printf("Condition is TRUE: a is greater than b AND c is not
            zero.\n");
    } else {
        printf("Condition is FALSE.\n");
    }

    printf("\nNow changing c to 0...\n");
    c = 0;
    printf("a = %d, b = %d, c = %d\n", a, b, c);

    if (a > b && c != 0) {
        printf("This line will not be printed.\n");
    } else {
        printf("Condition is now FALSE because c is zero.\n");
    }

    return 0;
}
```

Output

```
a = 15, b = 10, c = 5
Condition is TRUE: a is greater than b AND c is not zero.

Now changing c to 0...
a = 15, b = 10, c = 0
Condition is now FALSE because c is zero.


=== Code Execution Successful ===
```

13.

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int num = 12;

    printf("Checking if %d is divisible by both 2 and 3 (without
        modulus):\n", num);

    if ((num / 6) * 6 == num) {
        printf("%d is divisible by both 2 and 3.\n", num);
    } else {
        printf("%d is NOT divisible by both 2 and 3.\n", num);
    }

    printf("\nChecking if 10 is divisible by both 2 and 3:\n");
    num = 10;
    if ((num / 6) * 6 == num) {
        printf("%d is divisible by both 2 and 3.\n", num);
    } else {
        printf("%d is NOT divisible by both 2 and 3.\n", num);
    }

    return 0;
}
```

Output

```
Checking if 12 is divisible by both 2 and 3 (without modulus):
12 is divisible by both 2 and 3.

Checking if 10 is divisible by both 2 and 3:
10 is NOT divisible by both 2 and 3.


=== Code Execution Successful ===
```

14.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int x = 10, y = 20;
    printf("Before swap: x = %d, y = %d\n", x, y);

    // Swap using XOR
    x = x ^ y;
    y = x ^ y;
    x = x ^ y;

    printf("After swap:  x = %d, y = %d\n", x, y);

    return 0;
}
```

Output

```
Before swap: x = 10, y = 20
After swap:  x = 20, y = 10


=== Code Execution Successful ===
```

15.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int num = 12;

    printf("Checking if %d is both positive and even:\n", num);
    if (num > 0 && num % 2 == 0) {
        printf("%d is both positive and even.\n", num);
    } else {
        printf("%d is NOT both positive and even.\n", num);
    }

    printf("\nChecking if -7 is both positive and even:\n");
    num = -7;
    if (num > 0 && num % 2 == 0) {
        printf("%d is both positive and even.\n", num);
    } else {
        printf("%d is NOT both positive and even.\n", num);
    }

    printf("\nChecking if 15 is both positive and even:\n");
    num = 15;
    if (num > 0 && num % 2 == 0) {
        printf("%d is both positive and even.\n", num);
    } else {
        printf("%d is NOT both positive and even.\n", num);
    }

    return 0;
}
```

Output

```
Checking if 12 is both positive and even:
12 is both positive and even.

Checking if -7 is both positive and even:
-7 is NOT both positive and even.

Checking if 15 is both positive and even:
15 is NOT both positive and even.


=== Code Execution Successful ===
```

16.

main.c

```c
#include <stdio.h>

int main() {
    int x = 10, y = 15;
    float average;

    average = (x + y) / 2.0;

    printf("The average of %d and %d is: %.2f\n", x, y, average);

    return 0;
}
```

Output

```
The average of 10 and 15 is: 12.50


=== Code Execution Successful ===
```

17.

main.c

```c
#include <stdio.h>

int main() {
    char ch = 'G';

    printf("Checking if '%c' is an uppercase letter:\n", ch);
    if (ch >= 'A' && ch <= 'Z') {
        printf("'%c' is an uppercase letter.\n", ch);
    } else {
        printf("'%c' is NOT an uppercase letter.\n", ch);
    }

    printf("\nChecking if 'g' is an uppercase letter:\n");
    ch = 'g';
    if (ch >= 'A' && ch <= 'Z') {
        printf("'%c' is an uppercase letter.\n", ch);
    } else {
        printf("'%c' is NOT an uppercase letter.\n", ch);
    }

    return 0;
}
```

Output

```
Checking if 'G' is an uppercase letter:
'G' is an uppercase letter.

Checking if 'g' is an uppercase letter:
'g' is NOT an uppercase letter.


=== Code Execution Successful ===
```

18.

main.c

```c
#include <stdio.h>

int main() {
    int a = 2, b = 3, c = 4;
    int sum_of_squares;

    sum_of_squares = (a * a) + (b * b) + (c * c);

    printf("The sum of squares of %d, %d, and %d is: %d\n", a, b, c,
        sum_of_squares);
    printf("Calculation: (%d * %d) + (%d * %d) + (%d * %d) = %d + %d +
        %d = %d\n",
            a, a, b, b, c, c, a*a, b*b, c*c, sum_of_squares);

    return 0;
}
```

Output

```
The sum of squares of 2, 3, and 4 is: 29
Calculation: (2 * 2) + (3 * 3) + (4 * 4) = 4 + 9 + 16 = 29


=== Code Execution Successful ===
```

19.

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int a = 10, b = 10, c = 5;

    printf("a = %d, b = %d, c = %d\n", a, b, c);

    if (a == b && b != c) {
        printf("Condition is TRUE: a equals b, and b does not equal c.\n");
    } else {
        printf("Condition is FALSE.\n");
    }

    printf("\nNow changing c to 10...\n");
    c = 10;
    printf("a = %d, b = %d, c = %d\n", a, b, c);

    if (a == b && b != c) {
        printf("This line will not be printed.\n");
    } else {
        printf("Condition is now FALSE because b equals c.\n");
    }

    return 0;
}
```

Output

```
a = 10, b = 10, c = 5
Condition is TRUE: a equals b, and b does not equal c.

Now changing c to 10...
a = 10, b = 10, c = 10
Condition is now FALSE because b equals c.


=== Code Execution Successful ===
```

20.

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int num = 15;

    printf("Checking if %d is a multiple of 3 or 5:\n", num);
    if (num % 3 == 0 || num % 5 == 0) {
        printf("%d is a multiple of 3 or 5.\n", num);
    } else {
        printf("%d is NOT a multiple of 3 or 5.\n", num);
    }

    printf("\nChecking if 9 is a multiple of 3 or 5:\n");
    num = 9;
    if (num % 3 == 0 || num % 5 == 0) {
        printf("%d is a multiple of 3 or 5.\n", num);
    } else {
        printf("%d is NOT a multiple of 3 or 5.\n", num);
    }

    printf("\nChecking if 7 is a multiple of 3 or 5:\n");
    num = 7;
    if (num % 3 == 0 || num % 5 == 0) {
        printf("%d is a multiple of 3 or 5.\n", num);
    } else {
        printf("%d is NOT a multiple of 3 or 5.\n", num);
    }

    return 0;
}
```

Output

```
Checking if 15 is a multiple of 3 or 5:
15 is a multiple of 3 or 5.

Checking if 9 is a multiple of 3 or 5:
9 is a multiple of 3 or 5.

Checking if 7 is a multiple of 3 or 5:
7 is NOT a multiple of 3 or 5.


=== Code Execution Successful ===
```

21.

main.c — Share — Run

```c
#include <stdio.h>

int main() {
    int x = 10, y = 20, z = 30;
    int temp;

    printf("Before cyclic swap: x=%d, y=%d, z=%d\n", x, y, z);

    temp = z;
    z = y;
    y = x;
    x = temp;

    printf("After cyclic swap:  x=%d, y=%d, z=%d\n", x, y, z);
    printf("(x became y, y became z, z became x)\n");

    return 0;
}
```

Output

```
Before cyclic swap: x=10, y=20, z=30
After cyclic swap:  x=30, y=10, z=20
(x became y, y became z, z became x)


=== Code Execution Successful ===
```

22.

main.c — Share — Run

```c
#include <stdio.h>
#include <math.h>

int main() {
    double a = 10.5, b = 15.5;
    int result;

    result = (int)round(sqrt(a + b));

    printf("a = %.1f, b = %.1f\n", a, b);
    printf("Sum: %.1f\n", a + b);
    printf("Square root of sum: %.3f\n", sqrt(a + b));
    printf("Rounded to nearest integer: %d\n", result);

    return 0;
}
```

Output

```
a = 10.5, b = 15.5
Sum: 26.0
Square root of sum: 5.099
Rounded to nearest integer: 5


=== Code Execution Successful ===
```

23.

main.c

Share Run

```c
#include <stdio.h>

int main() {
    int num = 16;

    printf("Checking if %d is a power of 2:\n", num);
    if (num > 0 && (num & (num - 1)) == 0) {
        printf("%d is a power of 2.\n", num);
    } else {
        printf("%d is NOT a power of 2.\n", num);
    }

    printf("\nChecking if 12 is a power of 2:\n");
    num = 12;
    if (num > 0 && (num & (num - 1)) == 0) {
        printf("%d is a power of 2.\n", num);
    } else {
        printf("%d is NOT a power of 2.\n", num);
    }

    printf("\nChecking if 32 is a power of 2:\n");
    num = 32;
    if (num > 0 && (num & (num - 1)) == 0) {
        printf("%d is a power of 2.\n", num);
    } else {
        printf("%d is NOT a power of 2.\n", num);
    }

    return 0;
}
```

Output

```
Checking if 16 is a power of 2:
16 is a power of 2.

Checking if 12 is a power of 2:
12 is NOT a power of 2.

Checking if 32 is a power of 2:
32 is a power of 2.


=== Code Execution Successful ===
```

24.

main.c

Share Run

```c
#include <stdio.h>
#include <math.h>

int main() {
    int num = 49;
    int integer_sqrt = (int)sqrt(num);

    printf("Checking if %d is a perfect square:\n", num);
    if (integer_sqrt * integer_sqrt == num) {
        printf("%d is a perfect square (sqrt = %d).\n", num, integer_sqrt);
    } else {
        printf("%d is NOT a perfect square.\n", num);
    }

    printf("\nChecking if 50 is a perfect square:\n");
    num = 50;
    integer_sqrt = (int)sqrt(num);
    if (integer_sqrt * integer_sqrt == num) {
        printf("%d is a perfect square (sqrt = %d).\n", num, integer_sqrt);
    } else {
        printf("%d is NOT a perfect square.\n", num);
    }

    printf("\nChecking if 64 is a perfect square:\n");
    num = 64;
    integer_sqrt = (int)sqrt(num);
    if (integer_sqrt * integer_sqrt == num) {
        printf("%d is a perfect square (sqrt = %d).\n", num, integer_sqrt);
    } else {
        printf("%d is NOT a perfect square.\n", num);
    }

    return 0;
}
```

Output

```
Checking if 49 is a perfect square:
49 is a perfect square (sqrt = 7).

Checking if 50 is a perfect square:
50 is NOT a perfect square.

Checking if 64 is a perfect square:
64 is a perfect square (sqrt = 8).


=== Code Execution Successful ===
```