Basant chand

C.Programming

Basant Chand

Westcliff University

C programming

Nirajan Thakuri

February 6, 2026

Q1. What is the purpose of the main () function in a C program? Explain its significance.

The starting point of a C program is the main functions. When you execute a C program first the computer will search for the main () function and start to execute the code which is written within it.

The importance of the main () function in C is that of all the functions, it is a starting point of the program, that is, it will continuously run all through. It regulates the program execution, arranging instructions and calling of other functions where necessary. A C program cannot be run or compiled without the main () function. It also sets a value (which is normally 0) to indicate that the program has been completed successfully.

Q2. Explain the difference between a variable declaration and a variable initialization in C.

C programming has variable declaration and variable initialization, which are differently used. A variable declaration is the point in which you declare a variable by stating the data type of the variable and the name of the variable to tell the compiler to reserve memory to the variable.

However, in this state the variable does not always have the value assigned to it. Conversely, giving the variable some initial value, which is usually provided at the declaration time, is referred to as variable initialization. For that reason, int a; is a declaration, and int a = 10; is a declaration assignment.

Q3. Write a C program to display a personalized greeting message. (Should contain 'hello' or 'welcome' in the message)

#include <stdio.h>

int main () {
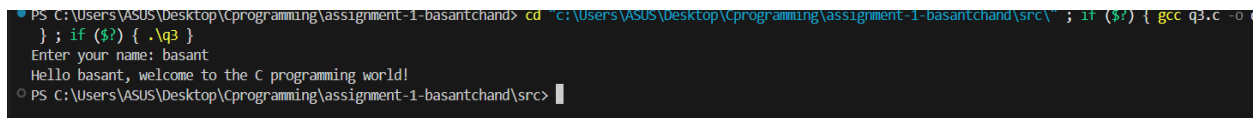
```
char name [50];

printf ("Enter your name: ");

 scanf ("%s", name);

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

return 0;

}
```

PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q3.c -o
 } ; if ($?) { .\q3 }
Enter your name: basant
Hello basant, welcome to the C programming world!
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>

Q4. What are the different data types available in C? Provide examples of each data type.

In C programming, data types define the type of data a variable can store. They help the compiler understand how much memory to allocate and how the data will be used. The main categories of data types in C are basic (primary), derived, and user-defined data types.

1.Basic (Primary) data types:

int – stores whole numbers

Example:

int age = 20;

float – stores decimal numbers (single precision)

Example:

float price = 99.5;

double – stores large decimal numbers (double precision)

Example:

double pi = 3.14159;

char – stores a single character

Example:

char grade = 'A';

void – represents no value (commonly used with functions)

Example:

void display();

2.Derived data types:

Array

int numbers[5] = {1, 2, 3, 4, 5};

Pointer

int *ptr;

Structure

struct Student {

 int id;

```
    char name[20];

};
```

3.Union

```
union Data {

    int i;

    float f;

};
```

User-defined data type:

Enum (Enumeration)

Enum Day {MON, TUE, WED};

Q5. Explain the concept of type conversions in C. Provide examples of implicit and explicit type conversions.

In C, type conversion is the conversion of a value of one type of data to another type. This normally occurs when an expression contains different types of data or when a computer programmer wishes to alter the type of a variable. Type conversion assists in making sure calculations and assignments are functional.

In C, the type conversion can be of two kinds; that is, implicit conversion and explicit conversion.

Automatic conversion (implicit type conversion) occurs automatically when the compiler is used to combine data types in an expression. The smaller type of data is normally translated to a larger type of data to prevent loss of data.

Example:

```
#include <stdio.h>

int main() {

    int a = 10;

    float b = 5.5;

    float result;

 result = a + b;  // int is automatically converted to float

    printf("Result = %f", result);

 return 0;

}
```

Explicit type conversion (type casting) is done manually by the programmer using a cast operator to convert one data type into another.

Example:

```
#include <stdio.h>

int main() {

    int a = 10, b = 3;
```

float result;

result = (float)a / b; // manual conversion

   printf("Result = %f", result);

return 0;

}

Q6. Write a C program to calculate the area of a rectangle. Prompt the user to enter the length and width, and display the result.

#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;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc
o q6 } ; if ($?) { .\q6 }
Enter the length of the rectangle: 3
Enter the width of the rectangle: 4
The area of the rectangle is: 12.00
```

Q7. What is the role of the scanf () function in C? Provide an example of its usage.

The scanf() function in C is used to take input from the user through the keyboard. It reads formatted data and stores it in variables. The function is defined in the stdio.h header file and uses format specifiers like %d, %f, and %c to read different types of values.

For example, %d is used for integers, %f for floating-point numbers, and %c for characters. When using scanf(), the address operator (&) is placed before variable names so the function can store the input value in memory.

Example program:

#include <stdio.h>

int main() {

   int age;

  printf("Enter your age: ");

   scanf("%d", &age);

  printf("You entered: %d", age);

 return 0;

}

In this example, scanf() reads an integer entered by the user and stores it in the variable age, which is then displayed using printf().

Q8. **Write a C program to convert temperature from Celsius to Fahrenheit. Prompt the user for a temperature in Celsius and display the equivalent temperature in Fahrenheit. (Formula: fahrenheit = (celsius * 9 / 5) + 32)**

```c
#include <stdio.h>

int main() {

    float celsius, fahrenheit;

    printf("Enter temperature in Celsius: ");

    scanf("%f", &celsius);

    fahrenheit = (celsius * 9 / 5) + 32;

    printf("Temperature in Fahrenheit: %.2f\n", fahrenheit);

    return 0;

}
```
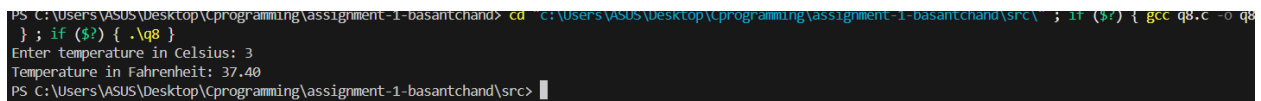
```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd  c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\  ; if ($?) { gcc q8.c -o q8
 } ; if ($?) { .\q8 }
Enter temperature in Celsius: 3
Temperature in Fahrenheit: 37.40
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q9. Input a number representing days and print the equivalent number of weeks and days. (Example: 10 days = 1 week and 3 days)

```c
#include <stdio.h>

int main() {

    int days, weeks, remainingDays;
```

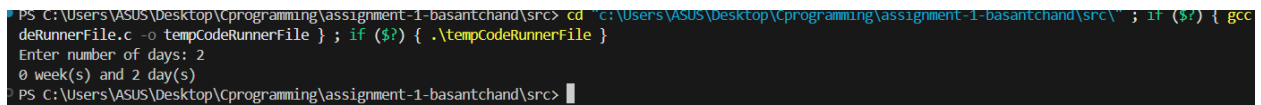printf("Enter number of days: ");

scanf("%d", &days);

weeks = days / 7;

remainingDays = days % 7;

printf("%d week(s) and %d day(s)\n", weeks, remainingDays);

return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\"; if ($?) { gcc
deRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number of days: 2
0 week(s) and 2 day(s)
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q10. Write a C program to swap the values of two variables using a temporary variable.

#include <stdio.h>

int main()

{

    int a, b, temp;

printf("Enter the value of a: ");

scanf("%d", &a);

printf("Enter the value of b: ");

scanf("%d", &b);

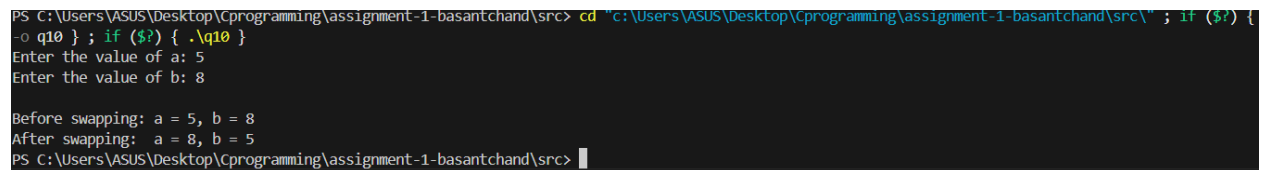printf("\nBefore swapping: a = %d, b = %d\n", a, b);

temp = a;

a = b;

b = temp;

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

return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) {
-o q10 } ; if ($?) { .\q10 }
Enter the value of a: 5
Enter the value of b: 8

Before swapping: a = 5, b = 8
After swapping:  a = 8, b = 5
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q11. Write a C expression that performs the following operations in a single line: increment a variable by 1, multiply it by 3, and subtract 10.

#include <stdio.h>

int main()

{

   int x = 5;

   int result;

```
    result = (++x * 3) - 10;

printf("Result: %d\n", result);

    printf("New value of x: %d\n", x);

return 0;

}
```

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?)
-o q11 } ; if ($?) { .\q11 }
Result: 8
New value of x: 6
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q12. Given three variables a, b, and c, write an expression that checks if a is greater than b and c is not equal to 0.

```
#include <stdio.h>

int main() {

    int a, b, c;

    printf("Enter value for a: ");

    scanf("%d", &a);

printf("Enter value for b: ");

    scanf("%d", &b);

printf("Enter value for c: ");

    scanf("%d", &c);

    if (a > b && c != 0) {
```

```
        printf("True\n");

    }

    else {

        // Fixed: It is 'else', not 'else5'

        printf("False\n");

    }

  return 0;

}
```
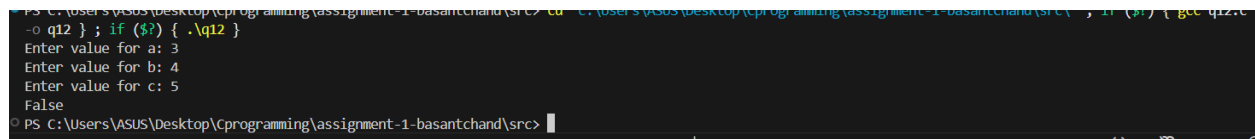
```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd   c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\ ; if ($?) { gcc q12.c
 -o q12 } ; if ($?) { .\q12 }
Enter value for a: 3
Enter value for b: 4
Enter value for c: 5
False
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q13. Write a C expression that evaluates whether a number is divisible by both 2 and 3 (without using the modulus operator).

```
#include <stdio.h>

int main()

{

    int num;

    printf("enter the number:");

    scanf("%d", &num);
```

int result;
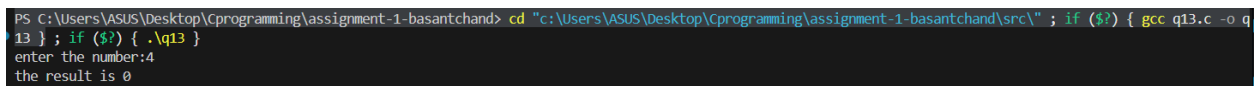
result= (num % 2== 0) && (num % 3==0);

printf("the result is %d\n", result);

return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q13.c -o q
13 } ; if ($?) { .\q13 }
enter the number:4
the result is 0
```

Q14. Create an expression that swaps the values of two variables x and y without using a temporary variable.

#include <stdio.h>

int main()

{

   int x, y;

   printf("enter the value of x and y:");

   scanf("%d %d", &x, &y);

 printf(" value of x before x is %d and y is %d", x, y);

   x = x + y;

   y = x - y;

x = x - y;

printf(" value of x is %d and y is %d\n", x, y);

return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q1
14 } ; if ($?) { .\q14 }
enter the value of x and y:2 3
 value of x before x is 2 and y is 3 value of x is 3 and y is 2
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q15. Write an expression that checks if a number is both positive and even.

#include <stdio.h>

int main() {

    int number;

    printf("Enter a number: ");

    scanf("%d", &number);

if (number > 0 && number % 2 == 0) {

        printf("The number is positive and even.\n");

    } else {

        printf("The number is NOT both positive and even.\n");

}

   return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q15.c
-o q15 } ; if ($?) { .\q15 }
Enter a number: 3
The number is NOT both positive and even.
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q16. Given two variables x and y, write an expression that calculates the average of their values.

  #include <stdio.h>

int main() {

   int x = 7;

   int y = 4;

 float average = (x + y) / 2.0;

    printf("x: %d\n", x);

   printf("y: %d\n", y);

   printf("Average: %.2f\n", average);

   return 0;

}

```
● PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q16.c -o q
  16 } ; if ($?) { .\q16 }
  x: 7
  y: 4
  Average: 5.50
● PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q17. Create an expression that checks if a given character is an uppercase letter.

#include <stdio.h>

int main()

{

    int result;

    char character;

    printf("enter the character: ");

    scanf("%c", &character);

    result = (character >= 'A') && (character <= 'Z');

    printf("the result is %d\n", result);

    return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q17.c
-o q17 } ; if ($?) { .\q17 }
enter the character: basu
the result is 0
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q18. Write a C expression that calculates the sum of the squares of three different numbers.

#include <stdio.h>

```c
int main() {

    int a = 2;

    int b = 3;

    int c = 5;

        int result = (a * a) + (b * b) + (c * c);

    printf("Sum of squares: %d", result);

    return 0;

}
```
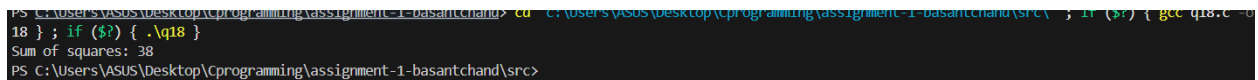


```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd   c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\   ; if ($?) { gcc q18.c -o
18 } ; if ($?) { .\q18 }
Sum of squares: 38
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q19. Given three variables a, b, and c, write an expression that checks if a is equal to b and b is not equal to c.

```c
#include <stdio.h>

int main() {

int x = 5;

    int math_result = (++x) * 3 - 10;

      printf("--- Part 1 ---\n");

    printf("Initial x: 5\n");
```

```c
    printf("Final x: %d\n", x);

    printf("Result ((++x) * 3 - 10): %d\n\n", math_result);

   int a = 10;

    int b = 10;

    int c = 99;

    int logic_result = (a == b) && (b != c);

    printf("--- Part 2 ---\n");

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

        if (logic_result) {

        printf("Result: TRUE (1) -> a equals b AND b is not c.\n");

    } else {

        printf("Result: FALSE (0)\n");

    }

   return 0;

}
```

```
-o q19 } ; if ($?) { .\q19 }
--- Part 1 ---
Initial x: 5
Final x: 6
Result ((++x) * 3 - 10): 8

--- Part 2 ---
a = 10, b = 10, c = 99
Result: TRUE (1) -> a equals b AND b is not c.
```

Q20. Write an expression that checks if a number is a multiple of either 3 or 5.

#include <stdio.h>

int main()

{

   int num;

   printf("Enter a number: ");

   scanf("%d", &num);

   if (num % 3 == 0 || num % 5 == 0)
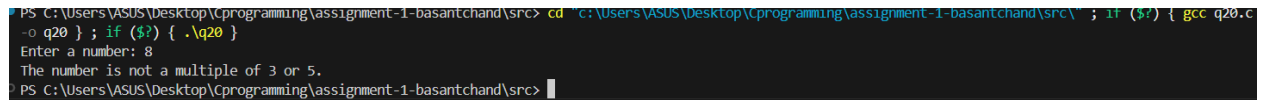
   {

     printf("The number is a multiple of 3 or 5.\n");

   }

   else

   {

     printf("The number is not a multiple of 3 or 5.\n");

}

return 0;

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q20.c
-o q20 } ; if ($?) { .\q20 }
Enter a number: 8
The number is not a multiple of 3 or 5.
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q21. Create an expression that swaps the values of three variables x, y, and z in a cyclic order (i.e., x becomes y, y becomes z, and z becomes x).

#include <stdio.h>

int main()

{

    int x, y, z, temp;

    printf("Enter x, y, z: ");

    scanf("%d %d %d", &x, &y, &z);

    temp = x;

    x = y;

    y = z;

    z = temp;

 printf("After cyclic swap:\n");

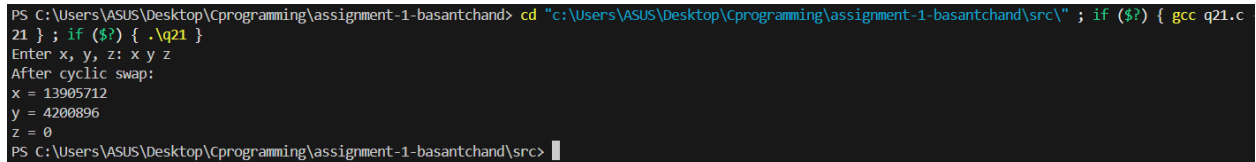   printf("x = %d\n", x);

```
printf("y = %d\n", y);

printf("z = %d\n", z);

return 0;
}
```

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q21.c
21 } ; if ($?) { .\q21 }
Enter x, y, z: x y z
After cyclic swap:
x = 13905712
y = 4200896
z = 0
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q22. Write a C expression that calculates the square root of the sum of two numbers, rounded to the nearest integer.

```
#include <stdio.h>

#include <math.h>

int main()

{

    double a, b, result;

    printf("Enter two numbers: ");

    scanf("%lf %lf", &a, &b);

    result = round(sqrt(a + b));
```
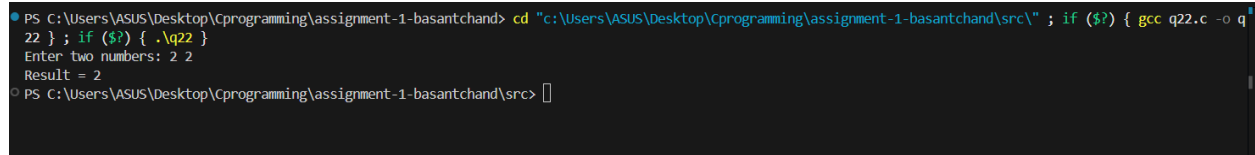
```
    printf("Result = %.0lf\n", result);

    return 0;

}
```

Q23. Given a variable num, write an expression that checks if it is a power of 2.

```
#include <stdio.h>

int main(){

    int num, result;

    printf("Enter the number: ");

    scanf("%d", &num);

    result = !(num & (num - 1));

    printf("The result is : %d\n", result);

    return 0;
```

}

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q23.c -o q
23 } ; if ($?) { .\q23 }
Enter the number: 4
The result is : 1
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```

Q24. Create an expression that checks if a given number is a perfect square.

#include <stdio.h>

#include <math.h>

int main()

{

   int num;

   int root;

   printf("Enter a number: ");

   scanf("%d", &num);

    root = (int)sqrt(num);

  if (num >= 0 && (root * root) == num)

  {

    printf("%d is a perfect square\n", num);

  }

  else

  {

```
        printf("%d is NOT a perfect square\n", num);

    }

    return 0;

}
```

```
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand> cd "c:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src\" ; if ($?) { gcc q24.c -o
24 } ; if ($?) { .\q24 }
Enter a number: 3
3 is NOT a perfect square
PS C:\Users\ASUS\Desktop\Cprogramming\assignment-1-basantchand\src>
```