

Name: Sujan Budha Chhetri

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

The main () function in a C program is the entry point of the program. Every C program must have exactly one main () function, because execution always starts from there and without this a program cannot run.

Its significance:

- **Entry point:** When the program runs, the operating system calls main (). Any code outside main () only executes if called from main () or other functions invoked from it.
- **Program control:** The main () provides a structured place to coordinate the program's flow. You can call other functions, handle input/output, and control program logic from here.
- **Return value to the operating system:** main () usually returns an integer (int).
 - i) return 0; indicates successful execution.
 - ii) Non-zero values indicate errors or abnormal termination.

This return value can be useful when running programs from scripts or other software that checks success or failure.

- **Defines program lifetime:** The program starts when main () begins and ends when main () finishes execution. Any code outside main () runs only if explicitly called.

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

Variable Declaration

- **Purpose:** It tells the compiler the type and the name of the variable.
- **Does not assign a value:** The variable may contain garbage data until it is assigned.
- **Syntax example:**

```
int age; // 'age' is declared as an integer
```

```
float salary; // 'salary' is declared as a float
```

Variable Initialization

- **Purpose:** It assigns an initial value to the variable at the time of declaration.
- **Syntax example:**

```
int age = 20; // declared and initialized
```

```
float salary = 5000; // declared and initialized
```

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

Basic Data Types

These are the fundamental types that are used to store simple values.

- **int**: It stores integers. For example: `int age = 25;`
- **float**: It stores single-precision decimal number. For example: `float price = 19.99;`
- **double**: It stores double-precision decimal numbers. For example: `double pi = 3.14159;`
- **char**: It stores a single character. For example: `char grade = 'A';`

2. Derived Data Types

These are built from basic types.

- **Array**: It stores multiple values of the same type. For example: `int numbers [5] = {1, 2, 3, 4, 5};`
- **Pointer**: It stores the address of a variable. For example:
 - `int *ptr;`
 - `int x = 10;`
 - `ptr = &x;`
- **Structure (struct)**: It groups different types of variables under one name. for example:

```
struct Point {
```

```
int x;
```

```
int y;
```

```
};
```

- **Union:** It is similar to struct but stores only one member at a time. For example:

```
union Data {
```

```
int i;
```

```
float f;
```

```
char c;
```

```
};
```

3. Void Type

- **Void:** It represents no value. It is often used in functions that do not return anything. For example:

```
void display () {
```

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

```
}
```

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

In C, type conversion refers to the process of converting a value from one data type to another. This is important because operations between different types may require the compiler

to convert values to a compatible type. Type conversions in C are classified into implicit (automatic) conversion and explicit (manual) conversion.

Implicit Type Conversion (Type Casting / Type Promotion)

- The compiler automatically converts a value from one type to another when needed.
- Commonly occurs in arithmetic operations when operands have different types.
- **Example:**

```
int a = 10;
```

```
float b = 5.5;
```

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

```
printf("Result = %.2f\n", result);
```

Explicit Type Conversion (Type Casting)

- The programmer manually converts a value from one type to another using casting.
- **Syntax:** (type)value
- **Example:**

```
float pi = 3.14;
```

```
int x = (int)pi; // explicitly convert float to int
```

```
printf("x = %d\n", x); // Output: 3
```

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

In C, the scanf() function is used to read input from the user via the keyboard and store it in variables. It is part of the <stdio.h> library. The function allows the program to interact with the user by taking input of various data types like integers, floats, characters, and strings.

Role of scanf():

1. Accepts user input from standard input (keyboard).
2. Stores the input in variables whose addresses are provided.
3. Supports multiple data types through format specifiers like %d, %f, %c, %s.

Example:

```
#include <stdio.h>

int main() {

    int age;

    float height;

    printf("Enter your age: ");

    scanf("%d", &age); // read integer input

    printf("Enter your height in meters: ");

    scanf("%f", &height); // read float input

    printf("Your age is %d and height is %.2f meters.\n", age, height);

    return 0; }
```