Chapter 02: Variables, Data Types and Operators

Learning Objectives

After completing this chapter, students will be able to:

- Understand variables and constants.
- Explain variable declaration and initialization.
- Identify different data types in C.
- Understand format specifiers.
- Use arithmetic, relational and logical operators.
- Apply assignment and increment/decrement operators.
- Understand operator precedence and type conversion.

Part A: Understand in Bangla

1. What is a Variable?

সহজ বাংলায়

মনে করো, তোমার কাছে একটি বাক্স আছে। বাক্সটির নাম `age`। এর মধ্যে তুমি ২০ সংখ্যাটি রাখলে।

Programming-এ এই বাক্সের মতো জায়গাকে Variable বলা হয়।

Variable হলো কম্পিউটারের Memory-তে তথ্য সংরক্ষণের জন্য ব্যবহৃত একটি নামযুক্ত জায়গা। Program চলার সময় এর value পরিবর্তন করা যায়।

English Exam Note

Definition:

A variable is a named memory location used to store data in a program. Its value can be changed during program execution.

Syntax

```
data_type variable_name;
```

Example

```
int age;
```

Here: - `int` is the data type. - `age` is the variable name.

2. Variable Declaration

বাংলায় ব্যাখ্যা

Variable Declaration হলো কোনো Variable-এর নাম এবং Data Type নির্ধারণ করে কম্পিউটারকে জানানো।

Example:

```
int age;
float salary;
char grade;
```

English Exam Note

Variable declaration is the process of specifying a variable's name and data type before using it in a program.

3. Variable Initialization

বাংলায় ব্যাখ্যা

Variable তৈরি করার সময়ই যখন তার মধ্যে কোনো প্রাথমিক মান (Initial Value) দেওয়া হয়, তখন তাকে Initialization বলে।

Example:

```c
int age = 20;
float height = 5.2;
char grade = 'A';
```

Declaration vs Initialization

Declaration	Initialization
Variable-এর নাম ও Type নির্ধারণ করে	Variable-এ প্রাথমিক Value দেয়
`int age;`	`int age = 20;`

4. Rules for Naming Variables

C programming-এ Variable-এর নাম দেওয়ার কিছু নিয়ম রয়েছে।

1. A variable name can contain letters, digits and underscores.
2. A variable name must not start with a digit.
3. Spaces are not allowed.
4. C keywords cannot be used as variable names.
5. Variable names are case-sensitive.
6. Special symbols such as @, # and % are not allowed.

Valid Variable Names

```
age
student_name
marks1
totalMarks
```

Invalid Variable Names

```
1age
student name
float
total-marks
```

মনে রাখবে: `age` এবং `Age` C programming-এ দুটি আলাদা Variable।

5. What is a Constant?

সহজ বাংলায়

Variable-এর Value পরিবর্তন করা যায়। কিন্তু Constant-এর Value Program চলার সময় পরিবর্তন করা যায় না।

Example:

```c
const float PI = 3.1416;
```

এখানে `PI` একটি Constant।

English Exam Note

A constant is a fixed value that cannot be changed during program execution.

Example

```c
#include <stdio.h>

int main()
{
    const int DAYS = 7;

    printf("%d", DAYS);

    return 0;
}
```

Output:

```
7
```

6. Data Types in C

সহজ বাংলায়

Variable-এর মধ্যে কী ধরনের তথ্য রাখা হবে, তা Data Type নির্ধারণ করে।

যেমন:

- পূর্ণ সংখ্যা: 25
- দশমিক সংখ্যা: 5.75
- একটি অক্ষর: 'A'

সব ধরনের তথ্য একইভাবে সংরক্ষণ করা যায় না। তাই C programming-এ বিভিন্ন Data Type রয়েছে।

English Exam Note

A data type specifies the type of data that a variable can store. It also determines the memory representation and operations applicable to that data.

Basic Data Types

Data Type	Description	Example
int	Integer numbers	25
float	Decimal numbers	5.75
double	Higher-precision decimal numbers	25.6789
char	Single character	'A'

Example

```c
int age = 20;
float price = 25.50;
double pi = 3.141592;
char grade = 'A';
```

Important: Exact memory size of a data type depends on the compiler and system. Use `sizeof()` to determine it.

Finding Memory Size

```c
#include <stdio.h>

int main()
{
    printf("int: %zu bytes\n", sizeof(int));
    printf("float: %zu bytes\n", sizeof(float));
    printf("double: %zu bytes\n", sizeof(double));
    printf("char: %zu byte\n", sizeof(char));

    return 0;
}
```

Output may vary depending on the system.

7. Format Specifiers

সহজ বাংলায়

`printf()` দিয়ে কোনো Variable-এর Value দেখাতে হলে তার Data Type অনুযায়ী Format Specifier ব্যবহার করতে হয়।

Important Format Specifiers

Format Specifier	Data Type
%d	int
%f	float

Format Specifier	Data Type
%lf	double in scanf
%c	char
%s	String
%u	unsigned int
%ld	long int
%zu	size_t

Note: In `printf()`, `%f` is used to print a floating-point value, including a `double`.

Example

```c
#include <stdio.h>

int main()
{
    int age = 20;
    float marks = 85.5;
    char grade = 'A';

    printf("Age = %d\n", age);
    printf("Marks = %.2f\n", marks);
    printf("Grade = %c\n", grade);

    return 0;
}
```

Output:

```
Age = 20
Marks = 85.50
Grade = A
```

Here, `%.2f` displays two digits after the decimal point.

8. Taking Input Using scanf()

বাংলায় ব্যাখ্যা

`scanf()` Function ব্যবহার করে User-এর কাছ থেকে Input নেওয়া যায়।

Example:

```c
#include <stdio.h>

int main()
{
    int age;

    printf("Enter your age: ");
    scanf("%d", &age);

    printf("Your age is %d", age);

    return 0;
}
```

Possible Input:

```
20
```

Output:

```
Enter your age: 20
Your age is 20
```

Important Point

- `printf()` displays output.
- `scanf()` takes input.
- `&age` represents the address of the variable `age`.
- The `&` operator is generally used with `scanf()` for ordinary variables.

---

9. Operators in C

সহজ বাংলায়

Operator হলো এমন একটি Symbol, যার সাহায্যে কোনো Operation বা কাজ সম্পন্ন করা হয়।

যেমন:

```
10 + 5
```

এখানে `+` হলো Operator। এটি দুটি সংখ্যাকে যোগ করে।

English Exam Note

An operator is a symbol that performs a specific operation on one or more operands.

The values on which an operator performs an operation are called operands.

Example:

```
a + b
```

Here: - `+` is the operator. - `a` and `b` are operands.

10. Types of Operators

C programming-এ গুরুত্বপূর্ণ Operators:

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Increment and Decrement Operators
6. Conditional Operator
7. Bitwise Operators

10.1 Arithmetic Operators

বাংলায় ব্যাখ্যা

Mathematical Calculation বা গাণিতিক হিসাব করার জন্য Arithmetic Operators ব্যবহার করা হয়।

Operator	Meaning	Example
+	Addition	10 + 5 = 15
-	Subtraction	10 - 5 = 5
*	Multiplication	10 * 5 = 50
/	Division	10 / 5 = 2
%	Modulus	10 % 3 = 1

Important: Modulus (%)

Modulus Operator ভাগশেষ বের করে।

Example:

```
10 % 3
```

এখানে ১০ কে ৩ দিয়ে ভাগ করলে ভাগশেষ ১।

তাই Output হবে:

```
1
```

Important: Integer Division

```
printf("%d", 10 / 3);
```

Output:

```
3
```

কারণ Integer Division-এ ভগ্নাংশের অংশ বাদ পড়ে।

অন্যদিকে:

```
printf("%.2f", 10.0 / 3);
```

Output:

```
3.33
```

10.2 Relational Operators

বাংলায় ব্যাখ্যা

দুটি Value-এর মধ্যে তুলনা করার জন্য Relational Operator ব্যবহার করা হয়।

এগুলো তুলনার ফলাফল হিসেবে True অথবা False প্রদান করে।

C programming-এ: - True = 1 - False = 0

Operator	Meaning	Example
==	Equal to	5 == 5
!=	Not equal to	5 != 3
>	Greater than	8 > 5
<	Less than	3 < 7
>=	Greater than or equal to	5 >= 5
<=	Less than or equal to	4 <= 6

Example

```c
#include <stdio.h>

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

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

    return 0;
}
```

Output:

```
1
0
```

Explanation: - 10 > 5 is True, so output is 1. - 10 == 5 is False, so output is 0.

10.3 Logical Operators

বাংলায় ব্যাখ্যা

একাধিক Condition একসঙ্গে পরীক্ষা করার জন্য Logical Operators ব্যবহার করা হয়।

C-তে তিনটি প্রধান Logical Operator রয়েছে।

Operator	Name	Meaning
&&	Logical AND	সব Condition True হলে True
||	Logical OR	যেকোনো একটি Condition True হলে True
!	Logical NOT	Result উল্টে দেয়

AND (&&)

দুটি Condition-ই True হলে Result True হবে।

A	B	A && B
1	1	1
1	0	0
0	1	0
0	0	0

OR (||)

যেকোনো একটি Condition True হলেই Result True হবে।

A	B	A || B
1	1	1
1	0	1
0	1	1
0	0	0

NOT (!)

Result-কে উল্টে দেয়।

A	!A
1	0
0	1

Example

```c
#include <stdio.h>

int main()
{
    int age = 20;

    printf("%d\n", age >= 18 && age <= 25);
    printf("%d\n", age < 18 || age > 60);
    printf("%d\n", !(age == 20));

    return 0;
}
```

Output:

```
1
0
0
```

10.4 Assignment Operators

বাংলায় ব্যাখ্যা

Variable-এর মধ্যে Value সংরক্ষণ বা Update করার জন্য Assignment Operator ব্যবহার করা হয়।

Operator	Example	Equivalent Expression
=	a = 5	a = 5
+=	a += 3	a = a + 3
-=	a -= 3	a = a - 3
*=	a *= 3	a = a * 3
/=	a /= 3	a = a / 3

Operator	Example	Equivalent Expression
%=	a %= 3	a = a % 3

Example

```c
#include <stdio.h>

int main()
{
    int a = 10;

    a += 5;

    printf("%d", a);

    return 0;
}
```

Output:

```
15
```

Explanation:

`a += 5` means `a = a + 5`.

10.5 Increment and Decrement Operators

বাংলায় ব্যাখ্যা

Increment Operator কোনো Variable-এর Value এক বাড়ায়।

Decrement Operator কোনো Variable-এর Value এক কমায়।

Operator	Meaning
++	Increase by 1

Operator	Meaning
--	Decrease by 1

Increment Example

```
int a = 5;
a++;
```

Now, `a = 6`.

Decrement Example

```
int b = 5;
b--;
```

Now, `b = 4`.

Pre-increment and Post-increment

Pre-increment: ++a

আগে Value বাড়ে, তারপর ব্যবহৃত হয়।

Post-increment: a++

আগে বর্তমান Value ব্যবহৃত হয়, তারপর Value বাড়ে।

Example:

```
#include <stdio.h>

int main()
{
    int a = 5;
    int b = 5;

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

    printf("%d\n", a);
```

```
    printf("%d\n", b);

    return 0;
}
```

Output:

```
6
5
6
6
```

Dry Run

Statement	a	b	Output
Initial Value	5	5	-
++a	6	5	6
b++	6	6	5
Print a	6	6	6
Print b	6	6	6

10.6 Conditional Operator

Conditional Operator হলো একটি Ternary Operator, যা তিনটি Operand নিয়ে কাজ করে।

Syntax:

```
condition ? expression1 : expression2;
```

Condition True হলে expression1 এবং False হলে expression2 execute হয়।

Example:

```c
#include <stdio.h>

int main()
{
    int age = 20;

    (age >= 18) ? printf("Adult") : printf("Minor");

    return 0;
}
```

Output:

```
Adult
```

10.7 Bitwise Operators

Bitwise Operators Binary Bits-এর ওপর কাজ করে।

Operator	Name
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
<<	Left Shift
>>	Right Shift

Example:

```c
#include <stdio.h>

int main()
{
```

```
    int a = 5;
    int b = 3;

    printf("%d", a & b);

    return 0;
}
```

Binary: - 5 = 0101 - 3 = 0011

AND Result:

```
0001 = 1
```

Output:

```
1
```

---

11. Operator Precedence

বাংলায় ব্যাখ্যা

একটি Expression-এ একাধিক Operator থাকলে কোন Operator আগে কাজ করবে, সেটি নির্ধারণ করে Operator Precedence।

Example:

```
int result = 10 + 5 * 2;
```

এখানে Multiplication আগে হবে।

তাই:

```
10 + 5 * 2
= 10 + 10
= 20
```

Common Operator Precedence

Priority	Operators	Description
1	()	Parentheses
2	++ – !	Unary operators
3	* / %	Multiplication, Division, Modulus
4	+ -	Addition, Subtraction
5	< <= > >=	Relational
6	== !=	Equality
7	&&	Logical AND
8	||	Logical OR
9	= += -= *= /=	Assignment

Note: Operators at the same precedence level generally follow their associativity rules. Parentheses can make expressions easier to understand.

Example

```c
#include <stdio.h>

int main()
{
    int result = 10 + 5 * 2;

    printf("%d", result);

    return 0;
}
```

Output:

```
20
```

---

12. Type Conversion

বাংলায় ব্যাখ্যা

একটি Data Type-এর Value অন্য Data Type-এ পরিবর্তন করাকে Type Conversion বলে।

C programming-এ Type Conversion প্রধানত দুই ধরনের।

1. Implicit Type Conversion

Compiler স্বয়ংক্রিয়ভাবে একটি Data Type-কে অন্য Data Type-এ পরিবর্তন করে।

Example:

```
int a = 10;
float b = a;
```

এখানে Integer Value 10 স্বয়ংক্রিয়ভাবে Float-এ রূপান্তরিত হয়েছে।

2. Explicit Type Conversion (Type Casting)

Programmer নিজে Type Conversion করে।

Syntax:

```
(data_type) expression;
```

Example:

```
#include <stdio.h>

int main()
{
    int a = 5;
    int b = 2;

    float result = (float)a / b;
```

```
    printf("%.2f", result);

    return 0;
}
```

Output:

```
2.50
```

Here, `(float)a` converts the integer value into a floating-point value before division.

Part B: English Exam Notes

Important Definitions

1. Variable

A variable is a named memory location used to store data in a program. Its value can be changed during program execution.

2. Constant

A constant is a fixed value that cannot be changed during program execution.

3. Data Type

A data type specifies the type of data that a variable can store and determines how that data is represented in memory.

4. Operator

An operator is a symbol that performs a specific operation on one or more operands.

5. Operand

An operand is a value or variable on which an operator performs an operation.

6. Type Conversion

Type conversion is the process of converting a value from one data type to another.

7. Operator Precedence

Operator precedence determines the order in which operators are evaluated in an expression.

Important Short Questions and Answers

Q1. What is a variable?

Ans: A variable is a named memory location used to store data whose value can change during program execution.

Q2. What is a constant?

Ans: A constant is a fixed value that cannot be changed during program execution.

Q3. What is the difference between declaration and initialization?

Ans: Declaration specifies the name and data type of a variable, whereas initialization assigns an initial value to it.

Q4. What are the basic data types in C?

Ans: The basic data types in C are int, char, float and double.

Q5. What is the purpose of printf()?

Ans: The printf() function is used to display formatted output on the screen.

Q6. What is the purpose of scanf()?

Ans: The scanf() function is used to read formatted input from the user.

Q7. What is a modulus operator?

Ans: The modulus operator (%) returns the remainder of an integer division.

Q8. What is the difference between = and ==?

Ans: The assignment operator (=) assigns a value to a variable, whereas the equality operator (==) compares two values.

Q9. What is the difference between ++a and a++?

Ans: ++a increments the value before it is used, whereas a++ uses the current value before incrementing it.

Q10. What is type casting?

Ans: Type casting is the explicit conversion of a value from one data type to another by the programmer.

Part C: Programming Examples

Program 1: Addition of Two Numbers

```c
#include <stdio.h>

int main()
{
    int a, b, sum;

    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);

    sum = a + b;

    printf("Sum = %d", sum);

    return 0;
}
```

Sample Input:

```
10 20
```

Output:

```
Sum = 30
```

Program 2: Calculate the Area of a Rectangle

Formula:

Area = Length × Width

```c
#include <stdio.h>

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

    printf("Enter length and width: ");
    scanf("%f %f", &length, &width);

    area = length * width;

    printf("Area = %.2f", area);

    return 0;
}
```

Sample Input:

```
5 4
```

Output:

```
Area = 20.00
```

Program 3: Find the Remainder

```c
#include <stdio.h>

int main()
{
    int a = 17;
    int b = 5;

    printf("Remainder = %d", a % b);

    return 0;
}
```

Output:

```
Remainder = 2
```

Program 4: Swap Two Numbers Using a Temporary Variable

```c
#include <stdio.h>

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

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

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

    return 0;
}
```

Output:

```
a = 20
b = 10
```

Explanation:

1. The value of a is stored in temp.
2. The value of b is assigned to a.
3. The original value of a is assigned to b.

Part D: Common Beginner Mistakes

Mistake	Correct Form
`int 1age;`	`int age1;`

Mistake	Correct Form
`int age = 20`	`int age = 20;`
`if (a = 5)`	`if (a == 5)` when comparison is intended
`char grade = "A";`	`char grade = 'A';`
`scanf("%d", age);`	`scanf("%d", &age);`
`10 / 3` expecting 3.33 as integer	Use floating-point division

Important Reminders

- C is a case-sensitive language.
- Every statement generally ends with a semicolon.
- A character is written inside single quotation marks.
- A string is written inside double quotation marks.
- Use the correct format specifier for each data type.
- Do not confuse assignment (=) with equality (==).

Part E: Practice Questions

Section A: Short Questions

1. Define a variable.
2. What is variable initialization?
3. Define a constant.
4. What is a data type?
5. Write four basic data types in C.

6. What is a format specifier?
7. What is the purpose of the modulus operator?
8. Differentiate between = and ==.
9. What are logical operators?
10. What is type casting?

Section B: Descriptive Questions

1. Explain variable declaration and initialization with examples.
2. Describe the basic data types in C.
3. Explain different types of operators in C with examples.
4. Differentiate between pre-increment and post-increment operators.
5. Explain operator precedence with a suitable example.
6. Discuss implicit and explicit type conversion.

Section C: Programming Questions

1. Write a C program to add two integers.
2. Write a C program to calculate the average of three numbers.
3. Write a C program to calculate the area of a rectangle.
4. Write a C program to find the remainder of two integers.
5. Write a C program to swap two numbers using a temporary variable.
6. Write a C program to convert an integer into a float using type casting.
7. Write a C program to calculate simple interest.

Chapter Summary

In this chapter, we learned:

- Variables and constants.
- Variable declaration and initialization.
- Rules for naming variables.
- Basic data types in C.
- Format specifiers.
- Input and output using scanf() and printf().
- Different types of operators.
- Operator precedence.
- Type conversion and type casting.

Remember: Variables store data, data types define the kind of data, and operators perform operations on that data.

End of Chapter 02

Next Chapter: Input, Output and Decision Making in C.