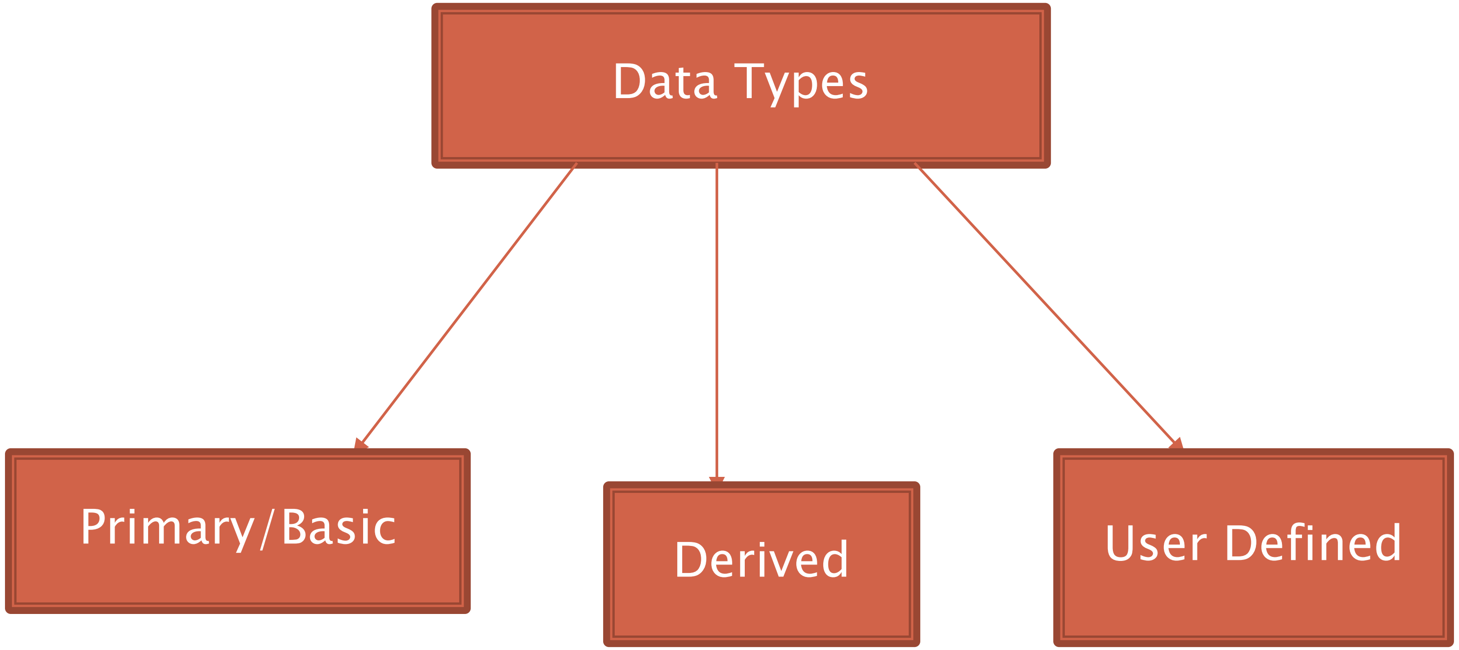
Data Types and Input Output - II

hitbullseye

Data Types

- Specifies the type of data, that can be stored in a variable in the form of integers, float values and characters. C language supports both signed and unsigned literals.



Primary/Basic Data Type

Includes int, float, char and double. The size of basic data types may vary according to 32 bit or 64 bit operating system.

Derived Data Type

Includes Arrays, Function and pointers.

User Defined Data Type

Includes Class, Structure and Union.

Memory Size of Data Types

- Char = 1 byte
- Integer = 2 bytes
- Float = 4 bytes
- Double = 8 bytes
- Long double = 10 bytes

- **Integer**- used to store the values without the decimal. The format specifier used for Integer is “%d”. e.g. int a=10;
- **Character**- used to store only single character. The format specifier used for character is “%c”. e.g. char ch=‘d’;
- **Floating-Point types-** used to store decimal values with single precision and exponential values. The format specifier used for floating point values is “%f”. e.g. float c=11.0
- **Double types-** used to store decimal values with double precision. The format specifier used for double data type is “%lf”. e.g. Double area =11.34

Input Output in C

- Input is to feed some data into a program.
- An input can be given in form of a file .
- Output is to display some data on any screen.
- C programming has standard input ,Standard output operations.

getchar() and putchar()

Int getchar()function is to reads the next character which is available on screen and returns it as an integer. It reads only single character at a time.

Int putchar() puts the passed character on screen and returns the same character.

Example:

```
#include<stdio.h>
int main()
{
int d;
printf(“enter a value”);
d = getchar();
Printf(“the value you entered”);
Putchar(d);
return 0;
}
```

Output:

```
enter a value: welcome to c
the value you entered : w
```

gets() and puts() functions

gets function reads a line from stdin.

puts function function writes the string.

Example:

```
#include<stdio.h>
int main()
{
char abc[100];
printf(“enter a value”);
gets(abc);
printf(“the value you entered”);
puts(abc);
return 0;
}
```

Output:

```
enter a value: welcome to c
the value you entered : welcome to c
```

printf() and scanf() functions

printf() reads the input from standard input stream stdin.

scanf()writes the output to standard output stream stdout.

Example:

```
#include<stdio.h>
int main()
{
char abc[100];
int d;
printf(“enter a value”);
scanf(“ %s  %d”, abc ,&d);
printf(“the value you entered : %s %d” ,str, d);
return 0;
}
```

Output:

```
enter a value: welcome  7
the value you entered : welcome 7
```

How to use data types in C?

```
#include<stdio.h> //header file for input output operations
#include<conio.h> //header file for console input output
void main()  //main function
{
Int a=20;
Char c=‘A’;
Float radius=7.00;
Double  area=6.78;
printf(“Value is :%d”, a);
printf(“Value of c : %c”, c);
printf(“Value of radius is: %f” ,f);
printf(“Value of area is : %lf” ,area);
getch();
}
```

Output:

```
Value is 20
Value of c :A
Value of radius is: 7.00
Value of area is : 6.78
```

Size of ()operator

- This operator is used to check the size of variable.

Example:

```
int main()
{
int c=sizeof(char);
int f=sizeof(float);
printf(“size of char : %c”, c);
printf(“size of float : %f”, f);
return 0;
}
```

Output:

```
size of char : 1
size of float: 4
```

Arrays

- An array is a variable that can store multiple values.
- It is a collection of elements of the same type placed in contiguous memory locations that can be individually referenced by using an index to a unique identifier.

e.g. If you want to store 50 integers , you create an array is int data[50];

Types of array:

Single Dimensional Array

Two Dimensional Array

Three Dimensional Array

Formatted input output

- Formatted input output are used to take various inputs from the user and display multiple outputs to the user.
- These types of I/O functions can helps to display the output to the user in different formats using the format specifiers.
- These I/O supports all types of data types e.g. int, float, char, and many more.

Format Specifies are:

S. No.	Specifier	Type	Explanation
1	%d	used for integers	used for integer value
2	%c	char	Used for I/O character value
3	%f	float	Used for I/O decimal floating-point value
4	%s	string	Used for I/O string/group of characters
5	%ld	long int	Used for I/O long signed integer value

Formatted I/O functions

- printf()- is used in a C program to display any value like float, integer, character, string, etc. on the console screen. It is declared in <stdio.h> header file.
- scanf()- used in the C program for reading or taking any value from the keyboard by the user, these values can be of any data type like integer, float, character, string. It is declared in <stdio.h> header file.

Formatted I/O functions

- sprintf()- stands for “string print”. This function is similar to printf() function but this function prints the string into a character array instead of printing it on the console screen.
- sscanf()- stands for “string scanf”. This function is similar to scanf() function but this function reads data from the string or character array instead of the console screen.

THANK YOU

hitbullseye