

ॐ



अयं बन्धुरयं नेति गणना लघुचेतसां उदारचरितानां तु वसुधैव कुटुम्बकं ॥

Upanishad's

Only small and narrow minded will think in terms of "this one is my relative and that one is a stranger",
but for noble and great hearts – **Entire World is a family, all are relatives.**

ॐ सर्वे भवन्तु सुखिनः।

सर्वे सन्तु निरामयाः।

सर्वे भद्राणि पश्यन्तु।

मा कश्चित् दुःख भाग्भवेत्॥

ॐ शान्तिः शान्तिः शान्तिः ॥

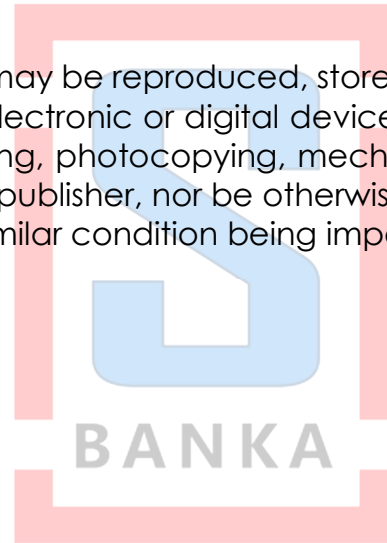
Meaning:

Om, May All Be Happy,
May all be Free from Illness,
May All See what is Auspicious,
May no one Suffer.
Om Peace, Peace, Peace.

Foundation of Computer Programming Using C, C++ & Java

Copyright © 2017, Subhash Banka (Author)

All rights reserved. No part of the publication may be reproduced, stored in a retrieval system (including but not limited to computers, disks, internal or external drives, electronic or digital devices, websites), or transmitted in any form or by any means (including but not limited to cyclostyling, photocopying, mechanical or electronic means and versions) without the prior written permission of the author and publisher, nor be otherwise circulated in any form of binding or cover other than that in which it is published and with a similar condition being imposed on the subsequent purchaser.



Banka

Computer

Education

www.sbanka.in

First Edition, First Print – 2017

Price : ₹750.00

ISBN No. : 978-93-5268-930-9

Published by : **AdiShiv Publications**
Patna 800023.
Mobile: 8294705101
Email: AdiShivpublications@gmail.com

Printed at : Vigyapan, Patna
0612 - 2521958



Foundation

of

Computer Programming

Using

C, C++ & Java

(First Edition)

Education

www.sbanka.in

Preface

I humbly present you with the book “Foundation of Computer Programming using C, C++ & Java”. The book tries to teach computer programming. This book is basically for people who are going to learn their first computer language, or who have already learnt one language and want to learn another language and for reference (but for programming basics). The ideal category would be college students in India.

The primary objective of the book is very simple “to Create a strong foundation”, which can be later used to learn more advanced and complex features of any language. The secondary objective that the book tries to achieve is comparative study of languages (here C, C++ & Java). This highlights the fact that you do not need to master ten languages but only one. In the rest of the languages you would always find that most of the concepts are same, the only difference will be in the approach or syntax (i.e. the way a command is given to the computer). And obviously different languages will provide some features unique to them but that would be a small part of the overall learning.

The entire book is in a narrative form. That is, I have not followed the usual pattern of text and then exercise but after discussing the essentials in a chapter, we directly move to programming and then as per the need and context, discuss the theory in between. This book uses the “programming first” approach to discuss programming as well as related theory or concept. The second thing you will miss in the book is a lot of text (theory) and objective or theory questions like define etc. The book has been designed purely for learning and I expect students to learn programming first and then try to evolve using various resources available to them. A lot of good material is available on the internet and there are many wonderful books written by experts and more knowledgeable people. In my opinion, providing everything in the book would have just increased the number of pages and would not have served the purpose of learning. The present generation has one of the biggest tool available for learning, Internet. Use it. No book can match the knowledge available there. This book just aims to prepare for more advanced learning by creating a solid foundation.

The idea for the book came in the July month of 2012 but the actual work started a year and a half later. The work progressed very slowly as I had to type & compose myself. Some other commitments for other books and the fact that since this was a comparative study, I had to take extra care to reconcile all the topics together from the three languages, made the progress even slower. All the programs in the book have been typed and tested in respective languages. The explanations are short and crisp and the complete program is on a single page (I made sure that none of the programs spreads over two or more pages). Five years later in June, 2017 the book was completed and I believe this book would make learning of computer programming easier.

Although I have tried to make sure that there are no mistakes in the book but my past experience with my other books as well as other writer’s book suggests that there would be errors in the book. My apologies. Please communicate any errors on my website www.sbanka.in. You are welcome to provide suggestions for any changes, additions or deletions in the book. You can rate the book. I would also look forward to any suggestions for the type of books the students need for my future work. As and if the book evolves and becomes popular, we would develop the website and add many features that would help in better learning.

I wanted to join Defense forces as a student but as the fate had ordained for me, I became a Computer Teacher at Don Bosco Academy in Patna. That was year 1992 and I had just completed my Nineteenth birthday. It was mentorship of Rajiv Sir and Don Bosco school management, which helped me evolve as a teacher. As with most people in this field in those days, I also wanted to be a programmer. In the year 1998, I finally decided to leave programming for teaching.

My sincere thanks to Mr. Rajiv Agarwal and the Don Bosco Family (the management, the teachers and the students) for helping me evolve with their help, trust, guidance and mentorship. There are so many people who have helped and guided me that naming a few would be injustice to them. A big THANK YOU to all who cared.

I need to thank all my students who have helped with this book by either giving suggestions, reading the draft of the book to find mistakes and encouraging me to write it.

Lastly I need to thank my family members who supported with me during the duration and especially my mother, Mrs. (Dr.) Urmila Banka, for allowing me to pursue a different career and helping with encouragement and blessing whenever needed and for working very hard to provide all comforts to me.

Subhash Banka



Spaceship

EARTH

Banka

Computer

Education

www.sbanka.in

Table of Contents

Part I

Chapter 1	The Bare Minimum Basic's (Input & Output)	Page No. 1
Chapter 2	Next Step: Taking Decisions (if...else)	Page No. 33
Chapter 3	Operators, Precedence & Associativity	Page No. 59
Chapter 4	Going in Circles (for loop)	Page No. 69
Chapter 5	More Circles (while & do...while loop)	Page No. 119
Chapter 6	Decisions, Once Again (switch...case)	Page No. 145
Chapter 7	Jump Statements (break & continue), Labels & goto, const & final, Escape Sequence, Tokens, Compound for loop, Chain Assignment, Short Circuiting	Page No. 161

Part II

Chapter 8	Functions/Methods	Page No. 171
Chapter 9	Array	Page No. 223
Chapter 10	String	Page No. 241
Chapter 11	2D Array	Page No. 289
Chapter 12	String Array	Page No. 309

PART III

Chapter 13	Searching & Sorting	Page No. 325
Chapter 14	User Defined Data Types	Page No. 343
Chapter 15	Object Oriented Programming	Page No. 353
Chapter 16	Pointers	Page No. 397
Chapter 17	Linked List	Page No. 409
Annexure I	Types of Errors in a Program	Page No. 440
Annexure II	Reserved Words	Page No. 441
Annexure III	Data Types	Page No. 442
Annexure IV	More on Input	Page No. 446
Annexure V	More on Printing	Page No. 447
Annexure VI	IDE (Integrated Development Environment)	Page No. 452

Part I



BANKA

Chapter 1

The bare minimum Basic's
(Input & Output)

Banka

Computer

Education

www.sbanka.in

What is a computer language?

Let's start with what is a language? Language is a means of communication which is effective only when both the parties trying to communicate know a common language.

Just like we have different languages (e.g. Hindi, French, German, English etc.) to communicate with each other, we have languages that we use to communicate with the computer. These languages are called computer languages e.g. BASIC, C, C++, Java etc.

A computer, being an electronic device can understand only absence (referred as 0) and presence (referred as 1) of electric current and hence a computer language consists of 0's and 1's only, and is known as Machine language or Binary language. But it is difficult for us to understand or write in 0's and 1's and hence to make our job simpler we created languages that use simple English language to give instructions to the computer. But the computer being a machine cannot understand English language directly and hence we need a translator. A **translator** converts our instructions given in English language to Machine understandable language of 0's and 1's which the computer can understand. **Compiler & Interpreter** are two types of translators.

Compiler and Interpreter are two special programs that convert (translates) a program written by us in any Computer language to machine language. But if two things perform same function then surely there must be some difference between the two (otherwise they will not be two).

To understand the difference between the two we shall take an example. Let's assume that a teacher is giving a lecture to a group of students in Japanese language. None of the students in the group understands Japanese language. Then how are they going to understand the teacher. The teacher has two options – he/she can take services of a person who will translate his/her lecture line by line to the audience as he/she speaks or the teacher can prepare the lecture in advance and get it converted to the local language beforehand and just distribute the photocopies of the converted lecture to the students. First approach is that of an Interpreter and the second is that of a Compiler.

Interpreter (BASIC, Java etc.)	Compiler (C, C++, Java etc.)
It converts program line by line to machine code and executes it immediately after conversion. (The person converting the lecture to students will not wait for the complete lecture to finish. He will keep translating it line by line as the teacher gives the lecture and then speak out the converted line to the audience, immediately. The process would repeat every time, the lecture is given)	It converts the entire program at a time but does not execute it immediately. (The person converting the prepared lecture will translate the entire speech as he is preparing a written version and this translated copy will be later distributed at the time of speech. If the same lecture has to be given again then we do not need to retranslate but use the last translation that was done).
It does not save the converted machine code (also known as object code).	It saves the converted machine code i.e. it creates the object code.
Since it converts line by line, it gives only one error at a time. The moment it encounters an error, program execution stops. Program has to be restarted after correcting the error.	Since it converts entire program at a time, obviously, it will produce a list of errors found in the entire code. These errors have to be removed before the actual object code can be created.
It is easier to remove one error at a time, hence, we say debugging is easier (debugging is a computer terminology to say removing of errors).	It produces number of errors, hence, debugging (removing of errors) tends to be comparatively more difficult.
Interpreter is a comparative simple software and working with it is slower.	Compiler is a more advanced software and compiled programs are faster. Since it works on entire program at a time, it is able to optimize the code for greater efficiency and speed.

What is a Program and what is programming?

Group of Instructions given to a Computer to achieve an objective is known as a **Program** and the process of writing these instructions (or a program) in any computer language can be termed as **Programming.**

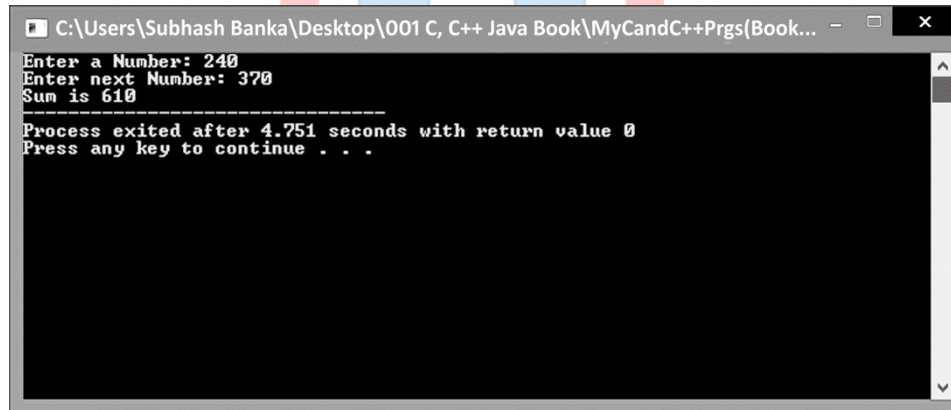
This book aims to help students learn not just the syntax (grammar) of a programming language but the actual coding in a computer language. It aims to develop logical thinking required to write programs, or to develop a more complex program using the simple ones learnt. This book aims to help create foundation, a beginner requires.

This book uses C, C++ and Java languages to explain programming concepts.

Programs are of various type. But as a beginner we shall deal with two simplest categorization:

- 1) **Character User Interface based (CUI).**
- 2) **Graphical User Interface based (GUI).**

1) **CUI** based programs are old style, dull to look at. They are basically screen with black background on which text is displayed in white. User has no flexibility but to follow steps as designed by the programmer. They are not interactive and are now outdated. In this style of program we use only keyboards to interact with the computer.



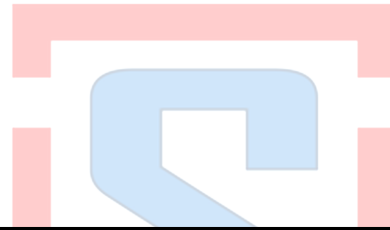
```
C:\Users\Subhash Banka\Desktop\001 C, C++ Java Book\MyCandC++Prgs(Book... - x
Enter a Number: 240
Enter next Number: 370
Sum is 610
-----
Process exited after 4.751 seconds with return value 0
Press any key to continue . . .
```

2) **GUI** based programs are more interesting and pleasing to look at. They are more colorful and interactive. The user decides the flow of program and uses mouse or touch to interact. They are quite easy to learn as well as handle. Today all computer programs developed are based on GUI interface.



It is my personal belief that to learn programming logic, nothing beats CUI. Once you gain sufficient confidence writing programs, then you should learn GUI based programming. The learning curve is greatly reduced and a student is fundamentally sounder with this approach. In this book we are dealing with only CUI based programming.

Anyways future and present is GUI based programming. So, once you finish this book it should be your first goal to learn GUI based programming.



INPUT & OUTPUT for Sample program on Next Page:

Example 1:

Enter a No. 23

Enter next No. 45

Sum is 68

Example 2:

Enter a No. 100

Enter next No. 235

Sum is 335

Example 3:

Enter a No. -45

Enter next No. -90

Sum is -135

NOTE

To learn programming there are multiple things that should be explained before we can even start our first line of the program and it is difficult to explain everything clearly in the beginning itself as a lot of things are based on concepts and topics that will be dealt later.

To make things simple in the following pages we shall be discussing all the essential features using one single program.

The first half of the page, displays the program, whereas in the second half, we shall be discussing about the things highlighted (in blue) in the program, one by one.

If everything is not clear in the first time, read again. Sometimes you will just need to move on and things will become clearer later.

Education

www.sbanka.in

C	C++	Java
<p>Major development work for C language was completed in early 70's. This language was initially designed to help creation of UNIX operating system. The credit for its development goes to Dennis Ritchie, who also co-designed UNIX OS.</p>	<p>In 80's Bjarne Stroustrup developed C++ as a superset of C language. The basic purpose of this language was to make programming simpler and to bring at one place some of the more advanced features at the time like classes. You do not need to know C to learn C++, but it helps.</p>	<p>Java was developed in early 90's by a team led by James Gosling. It was designed to take benefit of the Internet and provide cross platform compatibility. Java is an extension of Objective C (an Object oriented version of C) and some parts of Smalltalk and Simula68 (other languages of that time). It is one of the most popular languages used today. Knowledge of C or C++ is not necessary to learn Java, but it helps.</p>
<p>// A Sample program to add 2 no's in C</p> <pre>#include <stdio.h> void main(void) { int a, b, c ; printf ("Enter a No. "); scanf ("%d", &a) ; printf ("Enter next No. "); scanf ("%d", &b); c = a + b ; printf("Sum is %d" ,c) ; }</pre>	<p>// A Sample program to add 2 no's in C++</p> <pre>#include <iostream> using namespace std; int main(void) { int a, b, c ; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. "; cin >> b ; c = a + b ; cout << "Sum is " << c ; }</pre>	<p>// A Sample program to add 2 no's in Java</p> <pre>import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, c ; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); c = a + b ; System.out.println ("Sum is " + c); } }</pre>

Comments: The first line that is written in all the three formats is known as comments. **They are non-executable statements** i.e. they are not considered as part of your program. They are there just to **provide additional information about the program to anybody concerned**. For example, you write a program today and for some reason you do not need it for the next 2 years. After that period you again need to do some work on that program. Do you think you would remember everything about it at that time? The answer is No! Precisely for this reason we provide comments so that it can be referred at later date. It is also helpful when another programmer may want to study and understand your program and also to track changes and writer of code etc. **Comments can be placed anywhere in the program.**

There are two general ways to provide comment. It is same in all the three languages.

1) Single line comment (End-of-line comment) - Any text after the symbol // till the end of that line is treated as comment.

E.g. // From this point everything is all comment

2) Multi line comment (Block comment) - All the text that is written inside the block created by /* to */ is treated as comment.

E.g. /* This entire text between these two set of symbols is treated as comment */

Every line of the program written above is discussed one by one in the following pages.

It is a good programming habit to fill your programs with comments wherever possible or required. Although as a beginner I would suggest limit your comments to beginning and end of your programs.

C	C++	Java
<pre>// A Sample program to add 2 no's in C #include <stdio.h> void main(void) { int a, b, c ; printf ("Enter a No. "); scanf ("%d", &a); printf ("Enter next No. "); scanf ("%d", &b); c = a + b ; printf("Sum is %d" ,c); }</pre>	<pre>// A Sample program to add 2 no's in C++ #include <iostream> using namespace std; int main(void) { int a, b, c ; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. " ; cin >> b ; c = a + b ; cout << "Sum is " << c ; }</pre>	<pre>// A Sample program to add 2 no's in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, c ; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); c = a + b ; System.out.println ("Sum is " + c); } }</pre>

There are so many things that need to be explained even before you write your first program but that is what spoils the party. Too much theory makes the subject look boring and tough. But everything can't be avoided. So, I have tried to balance everything by discussing the non-essential topics in brief (and shall discuss later at the right time in detail). **Remember that as a beginner it is not necessary that you will understand everything in the first attempt. So, do not get disheartened if you face some problems. It is not important at what speed you are moving, what is more important that you keep moving.**

The line after comments tells the compiler that you are going to use a certain **header file** in your program (`#include . . .`). A header file contains declaration of **functions** that we may want to use in our programs. We may need to include more than one header files in our programs.

All C programs should always **#include <stdio.h>** and all C++ programs should **#include <iostream>**.

In older versions of C++ we include header file as in C, by adding .h in the declaration `<iostream.h>`. Newer versions of C++ also require **using namespace std;** line to indicate the use of standard (std) namespace for declarations.

Also in C++ we require to write **int main (void)** unlike in C where we can write **void main (void)**.

Many things expressed in the previous lines will appear confusing. But do not worry because **all these things have to be written in every program** and their meaning and utility is of no importance for now. They shall be and can be dealt later. **Most of the statements are terminated by a semi-colon (;) in C, C++ & Java (as you can see above)** and wherever used or required, you need to pay attention to the use of curly braces which is used to create a group of statements.

Actually, trying to deal with everything in the beginning will make things complicated. We shall deal with every topic before it is needed, at an appropriate time.

In Java **import** tells the java compiler about the **packages** we will be using in our program. Packages are not same but can be said to serve similar purpose as header files in C & C++. They contain **methods**. We may need to import more than one packages in our programs.

In a Java program, main () method should always have a declaration of **public static void main (String args []).**

Another thing is the **class Sample** block. Every Java program resides inside a class. In Java you also need to write **Scanner obj = new Scanner (System.in);** You can use any other name instead of obj in the line.

C	C++	Java
<pre>// A Sample program to add 2 no's in C #include <stdio.h> void main(void) { int a, b, c; printf ("Enter a No. "); scanf ("%d", &a); printf ("Enter next No. "); scanf ("%d", &b); c = a + b ; printf("Sum is %d" ,c); }</pre>	<pre>// A Sample program to add 2 no's in C++ #include <iostream> using namespace std; int main(void) { int a, b, c; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. "; cin >> b ; c = a + b ; cout << "Sum is " << c ; }</pre>	<pre>// A Sample program to add 2 no's in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, c; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); c = a + b ; System.out.println ("Sum is " + c); } }</pre>

Here we shall be discussing two important topics - Data types and variables. In the above example **int** is a data type and **a, b and c** are variables.

Any computer program basically works on values which can be number or names/text (names or text are called String in computer). These values are stored inside the computer's memory and which has an inbuilt mechanism to locate and use them as the need arises. **Variables** are names given to a memory location (a place inside the computer's memory) where data or values are stored. Since the content of this space can be changed as per our need it is known variable (something that varies or can vary). Once we give memory locations a name we just need to use the name given to use or to change the value stored in the location. Internal details of storing at that location and retrieving from that location is taken care by the computer automatically.

There are some **rules we need to follow to give variable their names:**

Data types are discussed on the next page.

- It cannot begin with any digit i.e. 0 to 9.
- It can contain only digits (0 to 9), Capital alphabets (A to Z), Small alphabets (a to z) and _ (underscore) characters. Java also allows the use of \$ (dollar) character in names. Capital and small letters are treated differently, so, variables having same name but different case are treated as different variables e.g. ABC, Abc, abc, aBC etc.
- It cannot be a **reserved word (or Keyword)**. A Reserved word (or Keyword) are already used and reserved by a computer language for its use. Hence, these words are not available to programmers for their use. E.g. if, else, import, int, for etc. **(List of reserved words in C, C++ & Java are given as an annexure at the end of this book).**

In practice variable names always use small letters. And a variable whose value will remain constant (i.e. will not change) in the program is named in capital letters. You should choose to follow these conventions. Use of any other characters other than the ones specified above will result in error. Do not ever use spaces or full stop (.) etc. in your names. Their use is a very common mistake done by many beginners. It is also highly advisable to give the variable names in context i.e. if you want to store age of a person then you can use "age" as a variable name or to store salary then the variable name can be "salary".

Try to use single word variable names but if you do need to use multiple words to name a variable, then keep the first letter of first word small and first letter of rest of the words capital like sumOfMarks, totalSalary, areaOfTriangle etc.

C	C++	Java
---	-----	------

In previous page we discussed that we store values in the computer memory and their location is given a name which we called as variable. The value that is stored in a variable can be of different types like numbers or text etc. and require different mechanism for data storage, that needs to be specified by giving the required data type.

Data Types: They specify the type of values that can be stored in a variable and amount of space required to store it. There are number of data type but we shall start with most basic of them - **int, float & double, char and String**.

int is used to store Integers (e.g. 50, 450, 7 etc.) whereas **float and double** is used to store decimal or fractional values (numbers with decimal point are also known as floating point numbers in computer terms e.g. 7.89, 400.50, 2.999 etc.). **char** is used to store a single character that can be typed e.g. the word Bat contains three characters. char types are enclosed within single quotes (e.g. 'B', 'A', 'T' etc.). **String** is zero or more characters enclosed within double quotes (e.g. "BAT", "A", "1st April", "" etc.).

Each of them will occupy different amount of memory space and will be able to store different amount of data. The space required by these data types and the range of values that they can store is given below. (Detailed discussion on data types is given as a separate annexure at the end of this book, page number for which is given at the last page of this chapter. Although the information given on this page is more than sufficient for a beginner).

Data Type	Size in Bytes	Lower Range	Upper Range	Data Type	Size in Bytes	Lower Range	Upper Range
int	2	-32768	32767	int	4	-2,147,483,648 (-2^{31})	2,147,483,647 ($2^{31}-1$)
float	4	1.2e-38 (1.2×10^{-38})	3.4e+38 (3.4×10^{38})	float	4	1.4e-45 (1.4×10^{-45})	3.4028235e38 ($\approx 3.4 \times 10^{38}$)
double	8	2.3e-308 (2.3×10^{-308})	1.7e+308 (1.7×10^{308})	double	8	4.9e-324 (4.9×10^{-324})	1.79769313486231e+308 ($\approx 1.8 \times 10^{308}$)
char	1	0	255	char	2	\u0000 (or 0)	\uffff (or 65535)

(\u0000 and \uffff are Unicode characters. More on this later)

The size and range of data types in both C and C++ is compiler and system dependent. So, they may be different in your machine. (Discussed later)

float and double data type perform similar actions, but there are some difference between the two. They are –

Float	Double
Comparatively requires less memory space to store data.	Comparatively requires more memory space to store data.
Stores Lower range of data.	Stores larger range of data.
It is also known as Single precision number (digits after decimal is less, up to 7).	It is also known as Double precision number (double digits after decimal, up to 16).
Since it occupies less space, working with this data type is faster, compared to double (If you want speed use float, but will have to compromise on range).	Since it occupies more space, working with this data type is comparatively slower (If you want range use double, but will have to compromise on speed).

Declaration, Assignment, Initialization: Every variable in C/C++/Java has to be declared and initialized before it can be used in a program.

Declaration means telling the compiler/interpreter the names and data types of the variables. When you write in the program, **int a, b, c;** we inform the compiler or interpreter that we want to use variables named a, b, c of int data type. Every variable needs to be declared before it can be used in a program.

Assignment is storing a value in a variable. E.g. a = 10 or c = 20.0, we can do assignment together with declaration also, e.g. int a=20, b, c=30; (Remember that in computer programming, data always moves from right side to the left side i.e. value on the right side is stored in the variable that is on the left side).

Initialization is the first assignment into a variable. Providing a starting value to a variable can also be termed as initialization. In C/C++ if we do not initialize then it provides a garbage value into our variables increasing the chance of error later. Java simply does not allow an uninitialized variable. It does not compile if it finds a variable is not initialized. **It is a good programming habit to always initialize your variables.**

This should suffice now. As of now everything is told on need to know basis. Any additional information will be provided as and when required.

C	C++	Java
<pre>// A Sample program to add 2 no's in C #include <stdio.h> void main(void) { int a, b, c ; printf ("Enter a No. "); scanf ("%d", &a); printf ("Enter next No. "); scanf ("%d", &b); c = a + b ; printf("Sum is %d" ,c) ; }</pre>	<pre>// A Sample program to add 2 no's in C++ #include <iostream> using namespace std; int main(void) { int a, b, c ; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. " ; cin >> b ; c = a + b ; cout << "Sum is " << c ; }</pre>	<pre>// A Sample program to add 2 no's in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, c ; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); c = a + b ; System.out.println ("Sum is " + c); } }</pre>

A program always works on certain values. **When while writing a program we do not have these values and we do not have any formulas to find the value required then these values need to be entered by the user of the program through the keyboard.** Every computer language provides a mechanism to accept data from the keyboard and so does C, C++ and Java. **Input** means taking data from the user through the keyboard (although input can be from various sources like mouse, Hard disk, Camera etc. but as a beginner we should concentrate on taking the input from the keyboard only).

We use scanf () in C programs to take the input from the keyboard. The scanf () function is written in different ways for different data types.

To input an int value in a variable a, we write –

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

"%d" indicates an int value and &a indicates that the value will be stored in **memory location of variable a**.

For other data types we use %f (for float), %lf (for double), %c (for char) and %s (for string).

We can also input more than one value together-

```
scanf ( "%d %f " , &a, &b ) ;
```

C++ uses cin for input. It is very easy and straight forward as it does not require different formats for different data types. In C++, data type is taken care by the compiler.

We simply write -

```
cin >> a ;
```

Where a is a variable and can be of any data type.

To input more than one values together, we write –

```
cin >> a >> b ;
```

In C & C++, string is a char array and the way it should be handled, is discussed in detail in the chapters dealing with strings, later.

Input process of Java is more sophisticated. We use Scanner class for this (classes are discussed later). When we write **Scanner obj = new Scanner (System.in);** we are creating an object of the class Scanner. Scanner class will be explained in more detail later. As of now we simply need to know the input mechanism. Scanner class contains many methods to help us input data. The list of methods is given for the basic data types discussed till now.

```
int    a = obj.nextInt();
float  a = obj.nextFloat();
double a = obj.nextDouble();

String a = obj.nextLine();

char   a = obj.next().charAt(0); OR
       a = obj.nextLine().charAt(0); ← this one is preferred
```

obj is the same variable as used in:
Scanner **obj** = new Scanner (System.in);

C	C++	Java
<pre>// A Sample program to add 2 no's in C #include <stdio.h> void main(void) { int a, b, c ; printf ("Enter a No. "); scanf ("%d", &a); printf ("Enter next No. "); scanf ("%d", &b); c = a + b ; printf("Sum is %d" ,c); }</pre>	<pre>// A Sample program to add 2 no's in C++ #include <iostream> using namespace std; int main(void) { int a, b, c ; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. "; cin >> b ; c = a + b ; cout << "Sum is " << c ; }</pre>	<pre>// A Sample program to add 2 no's in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, c ; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); c = a + b ; System.out.println ("Sum is " + c); } }</pre>

Next is Printing or Output. **Output** means display of result. The very purpose of writing a program is to produce output. Output can be printed on a printer or displayed on a screen. We are restricting ourselves to the screen of computer. Again, different languages provide different ways to do it.

printf () is used in C to display any output you would want on the screen of the computer.

```
printf ("Hello India");
```

This simply prints text inside double quotes.

```
printf ("Sum is %d" , sum );
```

If the value of sum is 100, this will print **Sum is 100**.

Everything within the "" is printed but to print the value of variables we write %d (for int) and the actual variable name, outside the "" quotes.

```
printf ("Sum of %d and %d is %d" , a, b, sum );
```

If the value of a is 10, b is 20 and sum is 30, this will print **Sum of 10 and 20 is 30**. You can see that in the message to be printed we just wrote %d for the int values and actual variables were outside the "" quotes.

We write %d (for int), %f (float), %lf (double), %c(char), %s (string) and %p (pointers).

cout is used to display output on screen in C++. In C++, data types are handled by the compiler.

```
cout << "Hello India";
```

This prints text inside double quotes.

```
cout << "Sum is " << sum;
```

If the value of sum is 100, this will print **Sum is 100**.

```
cout << "Sum of " << a << " and " << b << " is " << sum ;
```

If the value of a is 10, b is 20 and sum is 30, this will print **Sum of 10 and 20 is 30**.

Remember, that when we input or store values in variables then we do not store what the values represent i.e. their units. E.g. if a variable contains 12 (int a = 12), it could mean anything. It could mean 12 years or 12 Kg or 12 cows.

The meaning is known to the programmer who uses the values accordingly. We cannot even put commas (,) between digits in a number (as is the habit in day to day life).

To display output on screen java provides System.out.println () & System.out.print () methods. Explanation is similar to C & C++.

```
System.out.println ("Hello India");
```

```
System.out.println ("Sum is " + sum);
```

```
System.out.println ("Sum of " +a+ " and " + b + " is " + sum);
```

System.out.print() is used when after printing the content within the bracket we do not want to change the line, which also implies that the output of next print statement will be on the same line as that of first one.

System.out.println () is used when we want the line to change after printing the content within the bracket, which also implies that the output of the next print statement will be on the next line.

```
System.out.print( "Hello ");
```

```
System.out.println( "World ");
```

Output is: **Hello World**

```
System.out.println( "Hello ");
```

```
System.out.print( "World ");
```

Output is: **Hello World**

C	C++	Java
<pre>// A Sample program to add 2 no's in C #include <stdio.h> void main(void) { int a, b, c ; printf ("Enter a No. "); scanf ("%d", &a); printf ("Enter next No. "); scanf ("%d", &b); c = a + b; printf("Sum is %d" ,c); }</pre>	<pre>// A Sample program to add 2 no's in C++ #include <iostream> using namespace std; int main(void) { int a, b, c ; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. " ; cin >> b ; c = a + b; cout << "Sum is " << c ; }</pre>	<pre>// A Sample program to add 2 no's in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, c ; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); c = a + b; System.out.println ("Sum is " + c); } }</pre>

Operators are special symbols which perform a specific operation on values provided (also known as operands). For example + (plus) is an operator, which will give the sum of two numbers and - (minus) is an operator, which will give difference of two numbers.

Operands are values on which the operators work. E.g. in the statement $a + b$, + (plus) is an operator, to perform addition on the two values **a and b (operands)**.

Expressions are a combination of operator and operands which would result in a value. For example $5 + 8 - 3$ is an expression which works on the operands 5, 8 and 3 using operators + and -, resulting in a value 10.

Operators can be categorized as Mathematical, Relational and Logical. We shall deal with Relational and Logical operators later as we do not need them now. Mathematical operators perform arithmetic operations and the ones we need to know as of now are -

- 1) + (addition) 2) - (subtraction) 3) * (Multiplication) 4) / (Division) 5) % (Mod for Remainder)

We need to use the exact symbols as specified here. Some examples are given below for your help-

interest = $(p*r*t)/100$; area = $3.14*r*r$; $d = b*b - 4*a*c$; distance = speed*time ; $f = (u*v)/(u+v)$; average = $(a+b+c)/2$;

= (equals to) is known as **assignment operator** and is used to store value or result on the right hand side into the variable given on the left side. In the above examples the result of expressions on the right hand side will be stored in the variable on the left hand side which are interest, area, d, distance, f and average.

So, in our program $c = a + b$; means that the sum of value stored in variable a and variable b is being stored in another variable called sum.

This finishes the basic discussion of everything written in our program. This basic knowledge now enables us to start writing our own programs. If anything is not clear than read again and do not hesitate to discuss with friends or teachers.

C	C++	Java
---	-----	------

All the mathematical operators are simple to understand but we will devote some more time and effort with % operator which gives remainder.

8%3 gives 2	8%13 gives 8	8%8 gives 0
5%2 gives 1	-5%2 gives -1	5%-2 gives 1
-5%-2 gives -1	8.5 % 3 gives error	

It is similar in java as in C and C++ with a difference that in Java we can use % operator with float and double values also.

We need to keep following point in mind to understand the above:

- % operator gives the remainder of integer division (quotient has to be an integer).
- It cannot be used with float and double data types in C and C++, hence the error in the last example.
- A smaller number cannot be divided by a bigger number, hence the smaller number itself is the remainder.
- With negative numbers the sign of the output depends on the sign of the numerator. To get the correct value simply ignore the negative sign(s), get the value and add sign of the numerator to the output.

So, in Java	8.5 % 3	gives	2.5
	-8.5 % 3	gives	-2.5
	8.5 % -3	gives	2.5
	-8.5 % -3	gives	-2.5

Actually the remainder is calculated by using the following formula for $x \% y$:

$$x = (x / y) * y + (x \% y) \text{ which means that } \mathbf{number(x) = quotient(x/y) * divisor(y) + remainder(x \% y).}$$

(Note: In this formula just remember that quotient (x/y) will always has to be in int.)

e.g. $-5 \% 2$ i.e. $x = -5$ and $y = 2$ implies $-5 = (-5/2) * 2 + (-5\%2) \Rightarrow -5 = (-2) * 2 + (-5\%2) \Rightarrow -5 = -4 + (-5\%2) \Rightarrow (-5 + 4) = (5\%2)$ which gives -1 as the answer.

The discussion above is purely academic. You can choose to ignore it, if you desire as it does not impact our programming skills in any way.

Operator precedence: tells us the order in which a computer language will solve the operators. There are a number of different operators in computer languages and their execution order is very important. Keeping things simple I am discussing precedence of mathematical operators only. Rest will be discussed later.

- We always solve * / % before + -
- Between * / % whichever comes first is solved first.
- Thereafter, between + - whichever comes first is solved first, although with this two operators the precedence has no effect on final output.

Examples	Result	Because
5 - 2 + 3	6	- is solved before +
5 * 2 / 5	2	* is solved before /
5 / 5 * 2	2	/ is solved before *
10 / 2 * 5	25	/ is solved before *
5 + 10 % 2	5	% is solved before +, (10%2 = 0)
5 + 5 * 2	15	* is solved before +
10 + 20 + 30 / 2	45	/ is solved before +

Order of precedence is affected by the use of brackets which are always solved first, before any other mathematical operator and also provide more clarity in expressions.

Examples	Result	Because
5 - (2 + 3)	0	+ is solved before - as it is inside a bracket ().
(10 + 20 + 30) / 2	30	+ is solved before / as it is inside a bracket ().

C	C++	Java
<pre>int a = 10 ; double b = a ; printf ("%lf" , b) ;</pre>	<pre>int a = 10 ; double b = a ; cout << b ;</pre>	<pre>int a = 10 ; double b = a ; System.out.print (b) ;</pre>

All three programs written above will print 10.0 as output. Why? We had stored an integer but get the output with decimal point (which is not integer). This happened because all the three languages converted int value to double data type, i.e. data type of the value has been changed by the respective compilers automatically. This change in of data types of the values is known as **Casting**.

The range of numbers that can be stored by int is far less than that of float, similarly the range that can be stored by float is far less than that of double. Assuming all the three data types mentioned here as containers, we can safely say that int is the smallest container whereas double is the biggest container. Now, if you apply common sense then we can easily deduce that transferring content from a smaller container to bigger will never pose any problem but the reverse may be a problem, especially when the bigger container contains more content than that can fit into smaller container.

Languages would convert values from int to both float and double and float to double automatically, as they pose no problem. This automatic type conversion is known as **Implicit casting or Coercion or widening**. The reverse poses problems as there can be loss of data. Hence when we try to convert double to int or float, or float to int, the compilers will produce error message. These conversions are only allowed if specifically instructed by the programmer. This programmer-defined type conversion is known as **Explicit casting or type casting or narrowing**.

C and C++, convert data types with higher range to data types with lower range directly (implicitly), without any need to be specified by the programmer explicitly. Data loss would happen if a bigger value is being stored in a data type with smaller range. This is also cause of many errors later and we need to be careful.

```
int a = 10 ;
double b = a ;           // implicit casting
double b = 7.85 ;
int a = b ;              // a will contain 7, performed automatically
int a = (int) b ;       // explicit casting, a will contain 7 (provides more clarity)
float c = b ;           // c will contain 7.85, performed automatically
float c = (float) c ;   // explicit, c will contain 7.85 (provides more clarity, prevents errors)
```

In Java, we have to specify to the compiler explicitly to convert a bigger range data type to smaller range data type, it is never done automatically as in C and C++.

```
int a = 10 ;
double b = a ;           // implicit casting
double b = 7.85 ;
int a = b ;              // Gives ERROR
int a = (int) b ;       // explicit casting, a = 7
float c = b ;           // Gives ERROR
float c = (float) b ;   // explicit casting, c=7.85
```

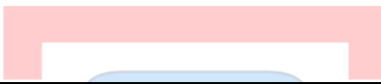
Important error that we need to be careful of: `float a = 50/100 * 200` // To find 50 % of 200, this should give 100.0 but gives 0.0

This statement looks correct but it will give wrong answer. In all the three languages mentioned above, data type of output always depends on the data type with highest range. Output of int with double will be double, int with float will be float, and float with double will be double. Similarly, output of double with double will be double, float with float will be float and int with int will be int. This is because of **numeric promotion**, which means automatic conversion of data type with lower range to the data type with higher range. The lower range data type will always be converted to the data type with highest range, out of all the data types involved in calculation.

So, in the above expression 50/100 (int/int) gives output in int i.e. instead of 0.5 it gives 0. And anything multiplied with 0 becomes 0. (Any number without decimal point is treated as int but a number with decimal point is treated as double in java, in C/C++ it does not matter as it will always take the data type of the variable being stored in (float or double). In Java a number is treated as float only if the number is followed by f as 45.75f or 45.75F or 80f or 80F etc. The above statement can be correctly written in either of the following ways-

- 1) `float a = 50f/100 * 200;`
- 2) `double a = 50.0/100 * 200;`
- 3) `float a = 50.0/100 * 200;`

- 1) `float a = 50f/100 * 200;`
- 2) `double a = 50.0/100 * 200;`
- 3) `float a = (float) (50.0/100 * 200);`



INPUT & OUTPUT for Question 1 on Next Page:

Example 1:

Enter a No. 10

Enter next No. 20

Sum is 30

Product is 200

Difference is -10

Example 2:

Enter a No. -5

Enter next No. -20

Sum is -25

Product is 100

Difference is 15

Example 3:

Enter a No. 20

Enter next No. -10

Sum is 10

Product is -200

Difference is 30

C	C++	Java
---	-----	------

1. Write a program to input two numbers and print their sum, product and difference.

```
// Answer to question 1 in C
#include <stdio.h>
void main( void )
{
    int a, b, sum, prod, diff ;

    printf( "Enter a No. " );
    scanf ( "%d", &a );

    printf( "Enter next No. ");
    scanf ( "%d", &b );

    sum = a + b ;
    prod = a * b ;
    diff = a - b ;

    printf( "Sum is %d \n" , sum ) ;
    printf( "Product is %d \n" , prod ) ;
    printf( "Difference is %d" , diff ) ;
}
```

In C lines do not change automatically. It can be changed as and when required by adding `\n` in the format string within "" double quotes.

```
// Answer to question 1 in C++
#include <iostream>
using namespace std;
int main( void )
{
    int a, b, sum, prod, diff ;

    cout << "Enter a No. " ;
    cin >> a ;

    cout << "Enter next No. " ;
    cin >> b ;

    sum = a + b ;
    prod = a * b ;
    diff = a - b ;

    cout << "Sum is " << sum << endl;
    cout << "Product is " << prod << "\n";
    cout << "Difference is " << diff ;
}
```

In C++ lines do not change automatically. It can be changed as and when required by adding `endl` at the end of cout. We can also use `\n`. Although I would be generally using `endl` in my programs.

```
// Answer to question 1 in Java
import java.util.Scanner ;
class Sample
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);

        int a, b, sum, prod, diff ;

        System.out.print ( "Enter a No. " );
        a = obj.nextInt();

        System.out.print ( "Enter next No. " );
        b = obj.nextInt();

        sum = a + b ;
        prod = a * b ;
        diff = a - b ;

        System.out.println ( "Sum is " + sum );
        System.out.println ( "Product is " + prod );
        System.out.println ( "Difference is " + diff );
    }
}
```

In Java, `println()` method **changes line after printing** the values enclosed in brackets. Java also allows use of `\n` like in C and C++ for change of line within the `print()` or `println()` methods.

I have highlighted the main part of the program. The top part and the closing curly braces at the bottom is something that you will have to write for every program. As a beginner my suggestion would be to first develop basic program logic and learn how to write, compile and execute your programs and then worry about meaning and significance of each and word, character or statement in the program.

Slowly as our confidence level increases we shall discuss in detail of various things that we are writing in our program and their importance. It is highly advisable to run your programs on computer and check them. It will help immensely to make you a good programmer. **The process of typing, compiling and executing programs in C, C++ and Java is given at the end of the book as separate Annexures (details about their page numbers is given on the last page of this chapter).**

Lastly, if something is not clear, do not give up. Read and analyze again and again. Discussion with friends and teachers always helps. There is always hope till you do not give up. Some people are able to catch things quickly whereas some take a bit more time. You need to work hard, with patience.



INPUT & OUTPUT for Question 2 on Next Page:

Example 1:

Enter a No. 5

Enter next No. 20

Result is -375

Example 2:

Enter a No. 20

Enter next No. -10

Result is 300

INPUT & OUTPUT for Question 3 on Next Page:

Example 1:

Enter a No. 10

Square is 100

Cube is 1000

Example 2:

Enter a No. 15

Square is 225

Cube is 3375

Education

www.sbanka.in

C	C++	Java
<p>2. Write a program to input two numbers and print product of their sum and difference.</p> <pre>// Answer to question 2 in C #include <stdio.h> void main(void) { int a, b, sum, prod, diff ; printf ("Enter a No. "); scanf ("%d", &a); printf ("Enter next No. "); scanf ("%d", &b); sum = a + b ; diff = a - b ; prod = sum * diff ; printf("Result is %d" , prod) ; }</pre>	<pre>// Answer to question 2 in C++ #include <iostream> using namespace std; int main(void) { int a, b, sum, prod, diff ; cout << "Enter a No. " ; cin >> a ; cout << "Enter next No. " ; cin >> b ; sum = a + b ; diff = a - b ; prod = sum * diff ; cout << "Result is " << prod ; }</pre>	<pre>// Answer to question 2 in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, b, sum, prod, diff ; System.out.print ("Enter a No. "); a = obj.nextInt(); System.out.print ("Enter next No. "); b = obj.nextInt(); sum = a + b ; diff = a - b ; prod = sum * diff ; System.out.println ("Result is " + prod); } }</pre>
<p>3. Write a program to input a number and print its square and cube.</p> <pre>// Answer to question 3 in C #include <stdio.h> void main(void) { int a, square, cube ; printf ("Enter a No. "); scanf ("%d", &a); square = a * a ; cube = a * a * a ; printf("Square is %d \n" , square) ; printf("Cube is %d" , cube) ; }</pre>	<pre>// Answer to question 3 in C++ #include <iostream> using namespace std; int main(void) { int a, square, cube ; cout << "Enter a No. " ; cin >> a ; square = a * a ; cube = a * a * a ; cout << "Square is " << square << endl; cout << "Cube is " << cube ; }</pre>	<pre>// Answer to question 3 in Java import java.util.Scanner ; class Sample { public static void main (String args[]) { Scanner obj = new Scanner (System.in); int a, square, cube ; System.out.print ("Enter a No. "); a = obj.nextInt(); square = a * a ; cube = a * a * a ; System.out.println ("Square is " + square); System.out.println ("Cube is " + cube); } }</pre>

INPUT & OUTPUT for Question 4 on Next Page:

C

Example 1:

Enter a No. 8

Square Root is 2.828427

Cube Root is 2.000000

Example 2:

Enter a No. 27

Square Root is 5.196152

Cube Root is 3.000000

Example 3:

Enter a No. 64

Square Root is 8.000000

Cube Root is 4.000000

C++

Example 1:

Enter a No. 8

Square Root is 2.82843

Cube Root is 2

Example 2:

Enter a No. 27

Square Root is 5.19615

Cube Root is 3

Example 3:

Enter a No. 64

Square Root is 8

Cube Root is 4

Java

Example 1:

Enter a No. 8

Square Root is 2.8284271247461903

Cube Root is 2.0

Example 2:

Enter a No. 27

Square Root is 5.196152422706632

Cube Root is 3.0

Example 3:

Enter a No. 64

Square Root is 8.0

Cube Root is 4.0

Between the three languages C, C++ & Java you find some variation while printing float and double values. That is normal and due to differences in their compilers.

C	C++	Java
---	-----	------

4. Write a program to input a number and print its square root & cube root.

```
// Answer to question 4 in C
#include <stdio.h>
#include <math.h>

void main( void )
{
    int a ;
    double sqroot, cuberoot ;

    printf ( "Enter a No. ");
    scanf ( "%d", &a );

    sqroot = sqrt (a);
    cuberoot = pow (a , 1.0/3);

    printf( "Square Root is %f\n" , sqroot);
    printf( "Cube Root is %f" , cuberoot);
}
```

sqrt(x) is a function in C to get the square root of a number x and pow (a, b) to find a to the power of b (a^b). It requires **math.h** header file to be included. We shall deal with functions in detail later.

Mathematically, Square root is same as power 1/2, cube root as power 1/3 and nth root as power 1/n. **Remember, int/int gives int, That is why, we need to careful while using this format and make either numerator or denominator as float or double.** A variable has to be declared as float or double whereas a number can be made float or double by adding a decimal part to it e.g. 1.5 or 1.0 etc.

e.g. $\sqrt[3]{x} = x^{1/3} = \text{pow}(x, 1.0/3)$
 $\sqrt[4]{x} = x^{1/4} = \text{pow}(x, 1.0/4)$
 $\sqrt[5]{x} = x^{1/5} = \text{pow}(x, 1.0/5)$
 $\sqrt[n]{x} = x^{1/n} = \text{pow}(x, 1.0/n)$

```
// Answer to question 4 in C++
#include <iostream>
#include <math.h>
using namespace std;
int main( void )
{
    int a ;
    double sqroot, cuberoot ;

    cout << "Enter a No. ";
    cin >> a;

    sqroot = sqrt (a);
    cuberoot = pow (a , 1.0/3);

    cout <<"Square Root is " << sqroot << endl;
    cout << "Cube Root is " << cuberoot;
}
```

sqrt(x) is a function in C++ to get the square root of a number x and pow (a, b) to find a to the power of b (a^b). It requires **math.h** header file to be included.

e.g. $\sqrt[3]{x} = x^{1/3} = \text{pow}(x, 1.0/3)$
 $\sqrt[4]{x} = x^{1/4} = \text{pow}(x, 1.0/4)$
 $\sqrt[5]{x} = x^{1/5} = \text{pow}(x, 1.0/5)$
 $\sqrt[n]{x} = x^{1/n} = \text{pow}(x, 1.0/n)$

```
// Answer to question 4 in Java
import java.util.Scanner ;
class Sample
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);

        int a ;
        double sqroot, cuberoot ;

        System.out.print ("Enter a No. ");
        a = obj.nextInt();

        sqroot = Math.sqrt (a);
        cuberoot = Math.cbrt (a); // Math.pow (a, 1.0/3)

        System.out.println ("Square Root is " + sqroot );
        System.out.println ("Cube Root is " + cuberoot );
    }
}
```

Java also provides Math.cbrt (a) method to find cube root. sqrt, cbrt, pow methods require **java.lang.Math** to be imported but it can be avoided as java compiler adds it by default (automatically). More on import later.

e.g. $\sqrt[3]{x} = x^{1/3} = \text{Math.pow}(x, 1.0/3)$ or $\text{Math.cbrt}(x)$
 $\sqrt[4]{x} = x^{1/4} = \text{Math.pow}(x, 1.0/4)$
 $\sqrt[5]{x} = x^{1/5} = \text{Math.pow}(x, 1.0/5)$
 $\sqrt[n]{x} = x^{1/n} = \text{Math.pow}(x, 1.0/n)$

If we write pow (x, 1/3) than it would imply pow (x, 0) which will always give 1 as anything to the power 0 is 1. In computer int/int=int. Because of which 1/3 means 0 and our statement became pow (x, 0), resulting in the above error.



INPUT & OUTPUT for Question 7 on Next Page:

C

Example 1:

Enter Number of shirts : 1
Amount to be paid is 437.500000

Example 2:

Enter Number of shirts : 5
Amount to be paid is 2187.500000

Example 3:

Enter Number of shirts : 10
Amount to be paid is 4375.000000

C++

Example 1:

Enter Number of shirts : 1
Amount to be paid is 437.5

Example 2:

Enter Number of shirts : 5
Amount to be paid is 2187.5

Example 3:

Enter Number of shirts : 10
Amount to be paid is 4375

Java

Example 1:

Enter Number of shirts : 1
Amount to be paid is 437.5

Example 2:

Enter Number of shirts : 5
Amount to be paid is 2187.5

Example 3:

Enter Number of shirts : 10
Amount to be paid is 4375.0

Between the three languages C, C++ & Java you find some variation while printing float and double values. That is normal and due to differences in their compilers.

Education

www.sbanka.in

C**C++****Java****Additional questions**

Attempt all the questions in the language of your choice. Attempt each and every question in the order given.

5. Write a Program to input the values of A & B and print the cube of their sum and cube of their product.

6. Write a Program to input two numbers and print the sum of 30% of each number.

7. Cost of each shirt in a shop is ₹ 500 only. The shop is giving a special festive discount of 12.5% on each shirt. Write a Program to input the number of shirts purchased and print the amount payable.

```
#include <stdio.h>
void main( void )
{
    int shirts ;
    double selling_price, amount ;

    printf( "Enter Number of shirts : " );
    scanf ( "%d", &shirts ) ;

    selling_price = 500 - (12.5/100*500);

    amount = shirts * selling_price ;

    printf("Amount to be paid is %f" ,amount);
}
```

```
#include <iostream>
using namespace std;
int main( void )
{
    int shirts ;
    double selling_price, amount ;

    cout << "Enter Number of shirts : " ;
    cin >> shirts ;

    selling_price = 500 - (12.5/100*500);

    amount = shirts * selling_price ;

    cout << "Amount to be paid is " << amount ;
}
```

```
import java.util.Scanner ;
class Book
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);

        int shirts;

        System.out.print ("Enter Number of shirts : ");
        shirts = obj.nextInt();

        double selling_price = 500 - (12.5/100*500);

        double amount = shirts * selling_price ;

        System.out.println ("Amount to be paid is " + amount );
    }
}
```

8. Area of a triangle is calculated using the formula given where a, b, c are sides of a triangle- $\text{area} = \sqrt{(s * (s - a) * (s - b) * (s - c))}$ where $s = \frac{a+b+c}{2}$, Write a program to enter sides of the triangle and print its area.

9. Your friend has three cows. Second cow gives 1.5 times the milk of first cow and third cow gives 1.5 times the milk of second cow. Write a Program to print the total amount of milk that will accumulate if the first cow gives 8 liters of milk.

10. Write a Program to print the amount to be paid to purchase N shirts at ₹ X each having a discount of Y% on each shirt. Hint: Refer Question 7

11. Write a Program to input two numbers and print the sum of X% of first and Y% of second number. Hint: you have to input four numbers.

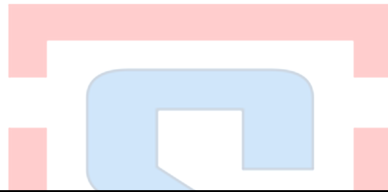
12. Write a Program to input the temperature in Centigrade (c) and print its equivalent in Fahrenheit (f). Where $f = ((9/5)*c) + 32$

13. Write a Program to input the temperature in Fahrenheit (f) and print its equivalent in Centigrade (c). Where $c = (5/9) * (f-32)$

14. Write a Program to enter the value of X and print the value of Y, where $y = \frac{x^6+x^4+x^2+10}{x^5+x^3+x}$

15. Write a Program to enter the value of X and print the value of Y, where $y = \frac{6\sqrt{x}+4\sqrt{x}+2\sqrt{x}+10}{5\sqrt{x}+3\sqrt{x}+x}$

Hint: Mathematically, Square root is same as power 1/2, cube root as power 1/3 and nth root as power 1/n (also remember that, int/int gives int).



INPUT & OUTPUT for Question 20 on Next Page:

C

Example 1:

Enter a Number 10

Result is 1.778279

Example 2:

Enter a Number 20

Result is 2.114743

Example 3:

Enter a Number 100

Result is 3.162278

C++

Example 1:

Enter a Number 10

Result is 1.77828

Example 2:

Enter a Number 20

Result is 2.11474

Example 3:

Enter a Number 100

Result is 3.16228

Java

Example 1:

Enter a Number 10

Result is 1.7782794100389234

Example 2:

Enter a Number 20

Result is 2.1147425268811277

Example 3:

Enter a Number 100

Result is 3.1622776601683786

Between the three languages C, C++ & Java you find some variation while printing float and double values. That is normal and due to differences in their compilers.

Education

www.sbanka.in

C	C++	Java
---	-----	------

16. Write a program to enter amount of purchase and print the amount payable if the shopkeeper gives a discount of 50% + 40% (This means that first you get a discount of 50% on the total value and then 40% on the remaining value that you have to pay after discount).

17. Write a program to enter amount of purchase and print the amount payable if the shopkeeper gives discount of 40% + 40% + 20%.

18. Write a Program to enter the values of u (object distance) & v (image distance) and print the value of f (focal length) where $\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$

19. Write a Program to input a number and print the cube of its square root.

20. Write a Program to input a numbers and print the cube of its square root of its cube root of its square root.

```
#include <stdio.h>
#include <math.h>
void main( void )
{
    int num ;
    double a , b , c , d ;

    printf( "Enter a Number " );
    scanf ( "%d", &num ) ;

    a = sqrt ( num ) ;
    b = pow ( a , 1.0/3 ) ;
    c = sqrt ( b ) ;
    d = c * c * c ;

    printf( "Result is %f" , d ) ;
}
```

```
#include <iostream>
#include <math.h>
using namespace std;
int main( void )
{
    int num ;

    cout << "Enter a Number " ;
    cin >> num ;

    double a = sqrt ( num ) ;
    double b = pow ( a , 1.0/3 ) ;
    double c = sqrt ( b ) ;
    double d = c * c * c ;

    cout << "Result is " << d ;
}
```

```
import java.util.Scanner ;
class Book
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);

        int num;

        System.out.print ( "Enter a Number " );
        num = obj.nextInt();

        double a = Math.sqrt ( num ) ;
        double b = Math.cbrt ( a ) ;
        double c = Math.sqrt ( b ) ;
        double d = c * c * c ;

        System.out.println ( "Result is " + d );
    }
}
```

In C & C++, we do not have functions to calculate cube root like we have in Java. Secondly in C, we have to declare variables before we actually use them unlike in C++ & Java, where we can declare them at the time of their first use.

21. A School conducts three term exams in a year. Final marks are calculated by giving different weightage to each term. First term has 10% weight, Second term has 20% weight and Final term has 70% weightage. Write a program to enter the marks of the three terms (each term of 900 marks) and print the final marks using weights given to each term.(Hint: Take 10% of marks scored in first term, 20% of second term, 70% of third term and print their sum).

22. Write a Program to input the Basic salary of a person and print his Take home pay. [Take home pay = Gross Pay – Deductions, HRA = 20% of (Basic salary + DA), DA = 98% of Basic salary, Gross Pay = Basic salary + HRA + DA, PF = 8.33% of (Basic salary + DA), IT = 20% of Gross Pay, Deductions = PF + IT.]

23. Write a program to enter number of 50, 25, 20, 10 and 5 paisa coins a person has and print the total value they would represent in rupees.

24. A circle inside the triangle touching all the three sides of the triangle is known as Inscribed circle and a circle outside the triangle touching all the three corners of the triangle is known as Circumscribed circle. Radius of circumscribed circle is calculated by $(a*b*c) / (4*AREA)$ and radius of Inscribed circle is calculated by $AREA / s$. Write a Program to print the area of the triangle, area of Inscribed and circumscribed circle.

25. A cow gives 15 liters of milk and eats 6 kg of fodder in a day. A farm produces 200 kg of fodder in a year. Write a Program to print the number of farms required, to feed the cows, required to produce milk for 10 Lakh people in a year, if each person drinks half liter of milk in a day. (Hint: Establish a link. Milk comes from cows, cows eat fodder and fodder comes from farms. So, first find total milk required, then number of cows required to give that amount of milk, from this you can get total fodder required every day and by multiplying by 365 you get the amount required in a year. And that will enable you to get farms required.)



INPUT & OUTPUT for Question 27 on Next Page:

C

Example 1:

Enter the Meter part of distance 10
Enter the centimeter part of distance 50
Distance in feet is 34.448819

Example 2:

Enter the Meter part of distance 20
Enter the centimeter part of distance 80
Distance in feet is 68.241470

C++

Example 1:

Enter the Meter part of distance : 10
Enter the centimeter part of distance : 50
Distance in feet is 34.4488

Example 2:

Enter the Meter part of distance : 20
Enter the centimeter part of distance : 80
Distance in feet is 68.2415

Java

Example 1:

Enter the Meter part of distance : 10
Enter the centimeter part of distance :50
Distance in feet is 34.44881889763779

Example 2:

Enter the Meter part of distance : 20
Enter the centimeter part of distance :80
Distance in feet is 68.24146981627297

Between the three languages C, C++ & Java you find some variation while printing float and double values. That is normal and due to differences in their compilers.

Education

www.sbanka.in

C	C++	Java
---	-----	------

26. Given that $a = 2*b$, $b = 5*c*d$, $c = 2*d*e$, $e = 7*d$. Write a program to input required values and print the final values of a, b, c, d & e.

27. Write a Program to input length in meters and centimeters and print its equivalent length in feet. (1 feet = 12 inches, 1 inch = 2.54 cm, 100 cm = 1 m).

```
#include <stdio.h>
void main( void )
{
    int meter, centimeter, totalcm ;
    double feet, inch ;

    printf("Enter the Meter part of distance");
    scanf ( "%d", &meter ) ;

    printf("Enter the centimeter part of distance");
    scanf ( "%d", &centimeter ) ;

    totalcm = (meter * 100) + centimeter ;
    inch = totalcm / 2.54 ;
    feet = inch / 12 ;

    printf("Distance in feet is %f", feet ) ;
}
```

```
#include <iostream>
using namespace std;
int main( void )
{
    int meter, centimeter, totalcm ;
    double feet, inch ;

    cout <<"Enter the Meter part of distance : ";
    cin >> meter ;

    cout<<"Enter the centimeter part of distance: ";
    cin >> centimeter ;

    totalcm = (meter * 100) + centimeter ;
    inch = totalcm / 2.54 ;
    feet = inch / 12 ;

    cout << "Distance in feet is " << feet ;
}
```

```
import java.util.Scanner ;
class Book
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);

        int meter, centimeter, totalcm ;
        double feet, inch ;

        System.out.print ("Enter the Meter part of
        distance : ");
        meter = obj.nextInt();

        System.out.print("Enter the centimeter part of
        distance :");
        centimeter = obj.nextInt();

        totalcm = (meter * 100) + centimeter ;
        inch = totalcm / 2.54 ;
        feet = inch / 12 ;

        System.out.println ("Distance in feet is " + feet ) ;
    }
}
```

Organizing the conversion formulas, we have:

- 1 feet = 12 inches
- 1 inch = 2.54 centimeters (cm)
- 100 cm = 1 meter (m)

To get distance in feet, we require the distance in inches. To get in inches we need the distance in centimeters. Therefore, we first convert the entire distance in meter and centimeter to centimeter, then to inches and at the end to feet.

28. Write a Program to input length in Feet and Inches and print its equivalent length in meters. (1 feet = 12 inches, 1 inch = 2.54 cm, 100 cm = 1 m).

29. Write a Program to input weight in Kilograms and grams and print its equivalent in Ounces. (1 pound (lbs.) = 16 ounces, 1 ounce = 28.35 grams, 1000 grams = 1 Kilogram).

30. Write a Program to input weight in Pounds (lbs.) and ounces and print its equivalent in Kilograms. (1 lbs. = 16 ounces, 1 ounce = 28.35 grams, 1000 grams = 1 Kilogram).

31. Write a Program to enter distance in Kilometer (Km), meter (m) and centimeter (cm) and convert it to millimeters (mm). (1 Kilometer = 1000 meter, 1 meter = 100 centimeter & 1 centimeter = 10 millimeter).

32. Write a Program to input Loan amount taken, yearly rate of interest at which the loan is taken and tenure of loan in months and to print the Equated monthly instalment (EMI) that will need to be paid every month to repay the loan in the specified tenure. Also print total interest that would be paid on the loan.

$$EMI = p * r * \frac{(1+r)^n}{((1+r)^n - 1)}$$

Where **p** is the loan amount, **n** is the tenure of loan (in months) and **r** is monthly rate of interest (= yearly rate of interest/100/12).

Total interest paid can be calculate by (EMI * n) - p.



OUTPUT for Question 35 on Next Page:

Sum of the series is 1230

There is no input in this program

Computer
Education

www.sbanka.in

C	C++	Java
---	-----	------

33. Write a Program to input Years, months, weeks and days elapsed since your birth (Basically your age, accurate up to today) and print its equivalent in hours, minutes. (1 year = 365 days, 1 month = 30 days, 1 week = 7 days, 1 days = 24 hours and 1 hour = 60 minutes).

34. Given that: $1 + 2 + 3 + 4 + \dots + n = \frac{n*(n+1)}{2}$, Write a Program to enter a number n and print the sum of all integers from 1 to n.

35. Given that: $1 + 2 + 3 + 4 + \dots + n = \frac{n*(n+1)}{2}$ Write a Program to determine the value of $10 + 11 + 12 + \dots + 50$.

```
#include <stdio.h>
void main( void )
{
    int n, a, b;

    n = 50 ;
    a = ( n*(n+1) )/2 ;

    n = 9 ;
    b = ( n*(n+1) )/2 ;

    printf("Sum of the series is %d", (a-b) );
}
```

```
#include <iostream>
using namespace std;
int main( void )
{
    int n = 50 ;
    int a = ( n*(n+1) )/2 ;

    n = 9 ;
    int b = ( n*(n+1) )/2 ;

    cout << "Sum of the series is" << (a-b) ;
}
```

```
class Book
{
    public static void main (String args[])
    {
        int n = 50 ;
        int a = ( n*(n+1) )/2 ;

        n = 9 ;
        int b = ( n*(n+1) )/2 ;

        System.out.println ("Sum of the series is"+(a-b) );
    }
}
```

Hint: Here we have subtracted the sum of numbers from 1 to 9 from the sum of numbers from 1 to 50. Resulting in sum of numbers from 10 to 50 as balance. Also note that in Java if we are not going to take input from keyboard we can choose not to write import java.util.Scanner; and Scanner obj = new Scanner (System.in).

36. Given that $1^2 + 2^2 + 3^2 + 4^2 + \dots + n^2 = \frac{n*(n+1)*(2*n+1)}{6}$, write a Program to determine the value of $8^2 + 9^2 + 10^2 + \dots + 100^2$.

37. Using the above formula write a program to print the result of following series: $(8^2 + 9^2 + 10^2 + \dots + 30^2) + (82^2) + (89^2 + 90^2 + \dots + 100^2) + (27^2 + 28^2 + \dots + 41^2)$

38. Write a Program to enter two numbers and Interchange (swap) their values. I am discussing two methods here, try to write the program yourself in language of your choice.

```
int a = 10, b = 20, temp ;
temp = a ;
a = b ;
b = temp ;
```

This method uses a third variable to interchange (swap) the values. The process could be understood with a simple example. Assume one glass A contains cold drink and another glass B contains Coffee. We need to interchange the glasses, how do we do it?

- 1) We shall take a third glass TEMP and pour cold drink from glass A into it (So now TEMP contains the cold drink).
- 2) Now we can pour Coffee from glass B into glass A (now glass A contains Coffee, not cold drink)
- 3) Lastly, we pour cold drink from glass TEMP to glass B (now glass B contains cold drink, completing our task).

```
int a = 10, b = 20 ;
a = a + b ;
b = a - b ;
a = a - b ;
```

This method swaps (interchanges) without using a third variable. This can be understood by using a simple mathematical fact:

If $a + b = \text{sum}$ then value of $a = \text{sum} - b$ and $b = \text{sum} - a$.

- 1) In first line, Sum of a & b is stored in a, i.e. $a = a + b$ (so a now represents sum).
- 2) Now, we want to store value of a in the variable b, but according to formula above value of $a = \text{sum} - b$. So for $b = a$, we use $b = \text{sum}$ (i.e. $a - b = a - b$ (as value of sum is stored in a). This stores value of a in to b.
- 3) Similarly, to store value of b into a, we write $a = \text{sum}$ (i.e. $a - a$ (i.e. $b = a - b$ (as value of sum is stored in a and value of a is stored in b). This now stores value of b in to a.

YOU MAY NEED TO READ & ANALYZE MORE THAN ONCE TO UNDERSTAND.



OUTPUT for Question 39 on Next Page:

Reverse is 21

There is no input in this program

Computer
Education

www.sbanka.in

C	C++	Java
---	-----	------

39. Write a Program to take a two digit number and print its digits in reverse order.

Hint: **When any integer is applied with % 10, then the result is always the last digit of that number.** Note: Remember % (modulus) operator given remainder.

Similarly, **When any integer is divided with 10, then the result is always removal of the last digit of that number.** Note: Remember int / int gives int.

```
12345 % 10 = 5
1234  % 10 = 4
123   % 10 = 3
12    % 10 = 2
1     % 10 = 1
```

```
12345 / 10 = 1234  same as divided by 10
1234  / 10 = 123   same as divided by 100
123   / 10 = 12    same as divided by 1000
12    / 10 = 1     same as divided by 10000
1     / 10 = 0     same as divided by 100000
```

You need to remember that in C, C++ and Java, **int/int gives an output in int.** This property has been used in the above examples to get the desired output.

- Firstly, we got the last digit as remainder when dividing by 10, because if we keep the quotient as integer then the last digit (with unit place value) will always be remainder. In Java, which allows use of non-integer values with % (modulus) operator, we need to make sure that we are dividing an integer value with 10.
- Secondly, because of the above mentioned property, we were able to remove that last digit when dividing by 10, as such division will always shift the digit at unit place, to after decimal in number, which will be removed as answer has to be in int.

```
#include <stdio.h>

void main( void )
{
    int n = 12, a, b, reverse ;

    a = n / 10 ;      // removes last digit
                    // giving the first digit

    b = n % 10 ;     // gives the last digit

    reverse = b * 10 + a ;

    printf( "Reverse is %d" , reverse ) ;
}
```

```
#include <iostream>
using namespace std;
int main( void )
{
    int n=12, a, b, reverse ;

    a = n / 10 ;      // removes last digit giving
                    // the first digit

    b = n % 10 ;     // gives the last digit

    reverse = b * 10 + a ;

    cout << "Reverse is " << reverse ;
}
```

```
class Program
{
    public static void main (String args[])
    {
        int n=12, a, b, reverse ;


        a = n / 10 ; // removes last digit giving the first digit

        b = n % 10 ; // gives the last digit

        reverse = b * 10 + a ;

        System.out.println ("Reverse is " + reverse );
    }
}
```

- 40.** Write a Program to take a three digit number and print it in reverse order.
- 41.** Write a Program to take a four digit number and print it in reverse order.
- 42.** Write a Program to enter an amount in rupees and print minimum number of currency of ₹2000, ₹500, ₹100, ₹50 and ₹1 that will be equivalent to the amount entered.
- 43.** Write a Program to enter a number and print the sum of digits at hundredth and thousandth position. (Assume that number entered is never less than 1000).
- 44.** Write a Program to input a three digit number and print the difference between the sum of digits of the number and product of digits of its reverse.
- 45.** Write a Program to input a four digit number (assume that correct number is entered) and print it after swapping the first digit with the last digit. For example if the input was 1234 then the output will be 4231.
- 46.** Write a Program to input number of years, months, weeks and days and print total number of days it represents. For example if we entered 5 years, 8 months, 2 weeks and 3 days, it would mean 2082 days (Take 1 year = 365 days, 1 month = 30 days and 1 week = 7 days).



INPUT & OUTPUT for Question 47 on Next Page:

Example 1:

Enter number of days 400

400 days equals 1 years 1 months 0 weeks 5 days

Example 2:

Enter number of days 1000

1000 days equals 2 years 9 months 0 weeks 0 days

Example 3:

Enter number of days 1500

1500 days equals 4 years 1 months 1 weeks 3 days

You will observe that despite using multiple print statements, output still remains on single line. That is because we have not changed line in any of our print statements used to give output. **None of the three languages, discussed change lines by themselves.**

C	C++	Java
---	-----	------

47. Write a Program to enter number of days and convert it to years, months, weeks & days (e.g. 452 days equals 1 year, 2 months, 3 weeks & 6 days). Assume that 1 year = 365 days & 1 month = 30 days and 1 week = 7 days.

```
#include <stdio.h>
void main( void )
{
    int days, weeks, months, years, rem_days ;

    printf( "Enter number of days " );
    scanf ( "%d", &days ) ;

    years = days / 365 ;
    rem_days = days % 365 ;    // remaining
                             // days

    months = rem_days / 30 ;
    rem_days = rem_days % 30; // remaining
                             // days

    weeks = rem_days / 7 ;

    rem_days = rem_days % 7 ; // remaining
                             // days

    printf( "%d days equals ", days ) ;
    printf( "%d years ", years ) ;
    printf( "%d months ", months ) ;
    printf( "%d weeks ", weeks ) ;
    printf( "%d days ", rem_days ) ;
}
```

```
#include <iostream>
using namespace std;
int main( void )
{
    int days, weeks, months, years, rem_days ;

    cout << "Enter number of days " ;
    cin >> days ;

    years = days / 365 ;
    rem_days = days % 365 ;    // remaining
                             // days

    months = rem_days / 30 ;
    rem_days = rem_days % 30 ; // remaining
                             // days

    weeks = rem_days / 7 ;

    rem_days = rem_days % 7 ; // remaining
                             // days

    cout << days << " days equals " ;
    cout << years << " years " ;
    cout << months << " months " ;
    cout << weeks << " weeks " ;
    cout << rem_days << " days " ;
}
```

```
import java.util.Scanner ;
class Book
{
    public static void main (String args[])
    {
        Scanner obj = new Scanner (System.in);
        int days, weeks, months, years, rem_days ;

        System.out.print("Enter number of days " ) ;
        days = obj.nextInt() ;

        years = days / 365 ;
        rem_days = days % 365 ;    // remaining
                                   // days

        months = rem_days / 30 ;
        rem_days = rem_days % 30 ; // remaining
                                   // days

        weeks = rem_days / 7 ;

        rem_days = rem_days % 7 ; // remaining
                                   // days

        System.out.print( days + " days equals " ) ;
        System.out.print( years + " years " ) ;
        System.out.print( months + " months " ) ;
        System.out.print( weeks + " weeks " ) ;
        System.out.print( rem_days + " days" ) ;
    }
}
```

Remember that [int/int gives an int](#). In this program we take advantage of that fact.

When numbers of days is divided by 365 we get years. E.g. for 400 days we get $400/365 = 1.096$ indicating more than a year but since we get the answer in int, we get 1.

Now when we apply % (mod, for remainder) on 400, we get $400\%365 = 35$ i.e. remaining 35 days out of 400 after subtracting for 1 year.

Similarly, we do our calculations for month, week and remaining days.

This finishes the first chapter. The beginning requires knowledge of many things that are actually based on the concepts or topics that are dealt at a later stage. We have tried to keep things as simple and minimum as possible, so as to tell you only the things required, without which program cannot be written.

One thing that you must have noticed is that all the programs were based directly or indirectly on Math. **Programming is based on Math**. All the problems that we solve through a program (whatever it might be), is actually ultimately broken down into mathematical steps. It wouldn't be wrong to say that "**Programming is conversion of mathematical steps into a computer language**". Although that is not a standard definition, but it highlights the importance of math and its understanding while writing a program. If you do not understand something mathematically or cannot break down the steps mathematically then you would find it difficult to write a program for it.

This book aims at developing these skills in you. Throughout the book we will try to understand the mathematical steps involved to solve a particular problem and we do hope that you would be able to link math with programming and become a good programmer or coder.

The following topics have been touched but not in details, which will be dealt in a later chapter. We are giving a list of all such topics and the page numbers at which they are discussed in detail (although as a beginner, in general, you should refrain from doing so).

	Topic	Page No. (in chapter 1)	Chapter No. (where discussed in more detail)	Page No.
1	Types of Errors in a Program		Annexure I	440
2	Header Files (C & C++)	6	8	219 - 220
3	using namespace std; (C++)	6	15	378 - 379
4	import (Java)	6	15	380
5	main ()	6	8	181 (point 5)
6	Reserved words	7	Annexure II	441
7	Data types	8	Annexure III	442
8	Unicode Character set	8	10	248
9	Declaration, Assignment & Initialization	8	3	67 - 68
10	More on Input	9	Annexure IV	446
11	More on printing	10	Annexure V	447
12	Operators - Mathematical, Relational, Logical	11	2	36, 60 - 61, 63 - 66
13	Operator Precedence	12	3	62
14	IDE (Integrated Development Environment) Writing and Compiling Programs in C & C++ Writing and Compiling Programs in Java	15	Annexure VI	452