Lecture – 02

Computer Programming

Marks Distribution:

Presentation	5%
Assignments	10%
Class Quizzes (2)	10%
Midterm Examination	25%
Final Examination	50%

Recommended Books:

1. C++ A Beginner’s Guide, Herbert Schildt, Tata McGraw Hill Edition.
2. Programming with C++, Second Edition, Dr. John R. Hubbard. Schaum’s Outline.

Contents

- Problem solving *(LL 04)*
- Six steps towards problem solution *(LL 04)*
- Basic problem solving concepts *(LL 04)*
  - Data types *(LL 04)*
  - Data literals*(LL 04)*
  - Variables (LL 04)
  - Constants (LL 04)
  - Rules for naming variable and constant (LL 04)
  - Operators (LL 02)

LL 02 = Learning Level 02 – Understanding, LL 04 = Learning Level 04 – Analysis

Problem Solving

Problem Solving

- In every day life people make decisions to solve many problems.

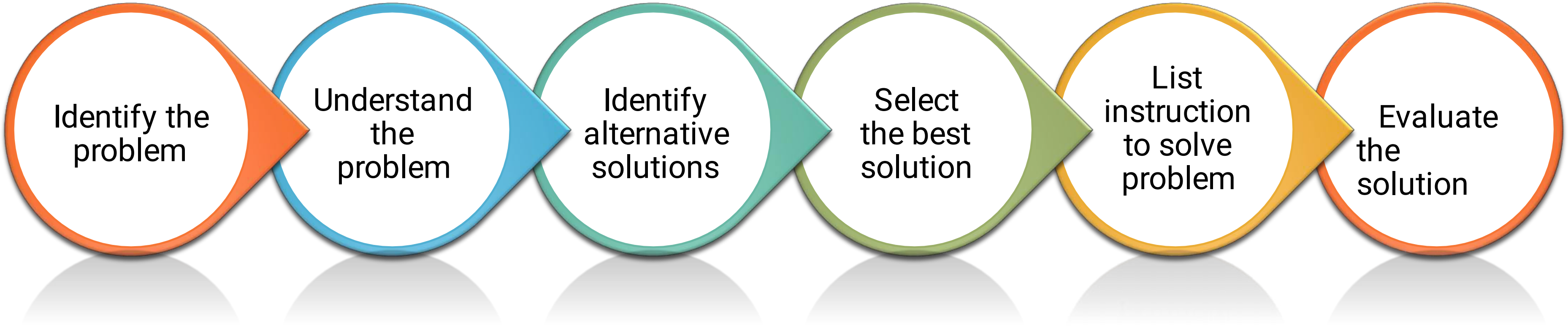
- The problems may be unimportant as what to watch on TV or as important
as choosing a new profession.

- If a bad decision is made, time and resources are wasted.

- So it is important that people know how to make decisions well.

Six Steps Towards Problem Solution

- There are six steps to follow to ensure the best decision:



Identify the problem

Understand the problem

Identify alternative solutions

Select the best solution

List instruction to solve problem

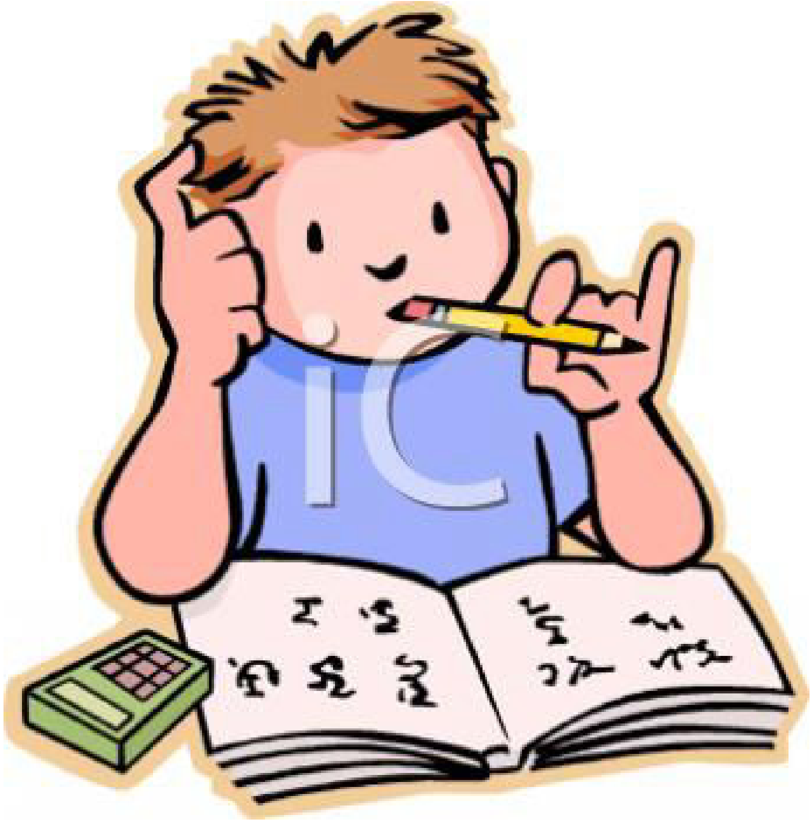
Evaluate the solution

Step 01: Identify the problem

Step 01: Identify the problem

- The first step towards solving a problem is to identify the problem.
- If you don’t know what the problem is, you cannot solve it.
- In a classroom situation, most problems have been identified for you and given to you in the form of written assignments or problems out of a book. However, when you are doing problem solving outside the classroom, you need to make sure you identify the problem before you start solving it.

Step 02: Understand the problem



Step 02: Understand the problem

- You must understand what is involved in the problem before you can continue toward the solution.
- You cannot solve a problem if you do not know the subject.
- For example, to solve a problem involving average of numbers, you must know how to calculate average; to solve a problem of trigonometry, you must know trigonometry.

Step 03: Identify alternative solutions

Step 03: Identify alternative solutions

- A single problem can be solved in many different ways.
- Identify and list all the possible solutions to the problem.
- For example there are multiple ways to sort the numbers as:
  - Insertion sort
  - Selection sort
  - Bubble sort

Step 04: Select the best solution

Step 04: Select the best solution

- Analyze and identify the pros and cons of every alternative solution.
- Choose the solution that is optimal and best matches with your requirements.
- For example out of all the sorting options you choose bubble sort as your best solution.

Step 05: List instructions to solve problem

Step 05: List instructions to solve problem

- Write down general step by step procedure to solve the problem.

- For example to solve problem of calculating average of three numbers:
  - Step 01 : Ask numbers *a*, *b*, and *c* from the user
  - Step 02 : Add *a*, *b*, *c* and store result in *sum*
  - Step 03 : Divide *sum* by 3 and store result in *avg*
  - Step 04 : Display *avg*

Step 06: Evaluate the solution

Step 06: Evaluate the solution

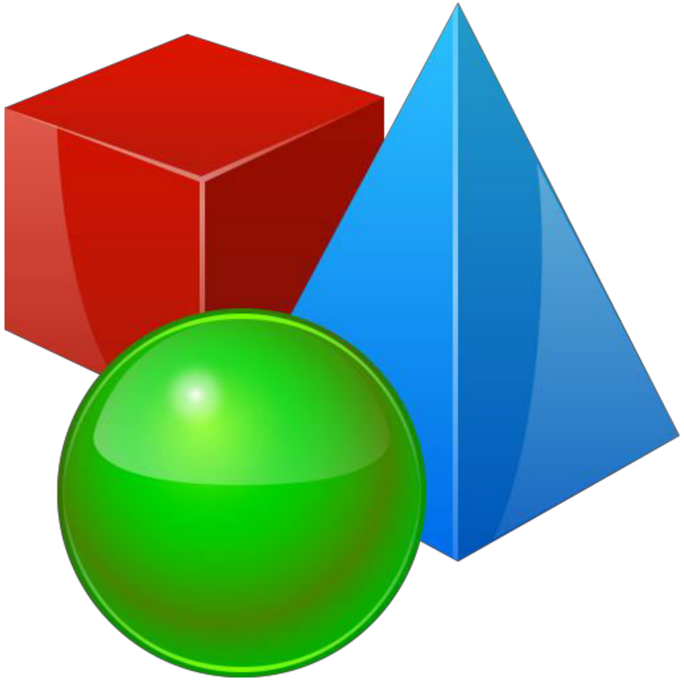
- Finally evaluate and test the solution, means to check its result to see if it is correct and satisfies the needs.

- For example, when a person needs a pen to write a letter, buying him marker may be a correct solution, but it may not be very satisfactory.

- If the result is either incorrect or unsatisfactory, then the problem solver must review the list of instructions to see that they are correct or start the process all over again.

Basic Problem Solving Concepts

Data Types



Data Types

- Every problem involves some sort of data in it.
- The data can be:
  - Numbers
- Integers (e.g. 24, 5874, -547)
- Floating (e.g. 45.214, 0.2547, -658.748)
- Characters (e.g. ‘a’, ‘f’, ‘#’, ‘?’, ‘!’, ‘w’)
  - String (e.g. “mehran”, “computer”, “MUET CSE”)
  - Logical (e.g. True and False, Yes and No, 0 and 1)
- The type of data is known as data type of that data item.

Data Types

Data Item	Example Value	Data Type
Age of a person	35	Integer
Current year	2014	Integer
Radius of circle	27.58	Floating
Value of PI	3.14159	Floating
A vowel	e	Character
A key on keyboard	?	Character
Person’s name	Ali Asghar	String
Address of an office	Department of CSE-MUET	String
Is 1st number greater than 2nd ?	True or T or Yes or 1	Logical
Is 7 less than 5 ?	False or F or No or 0	Logical

Do it yourself

Data

Data Item	Example Value	Data Type
Number of students		
Brand name of smartphone		
Is 5 an even number?		
Count of cars in parking		
Speed of a car		
Your grade in result		
Is 13 a prime number ?		
Title of book chapter		
Percentage of marks		
Option in MCQ		

Data Literals

- A fixed value that any data type can take is called as literal.
- A number literal is always written without quotes. (15, 68, 25.14, 578.14)
- A character literal is always written in single quotes. (‘a’, ‘f’, ‘#’, ‘?’, ‘!’)
- An string literal is always written in double quotes. (“mehran”, “computer”)
- An logical literal is always written without quotes. (True, False, T, F, Yes, No, 1, 0)

Data Literals

Data Literal	Type of Literal
“Ali”	String
‘b’	Character
25.2	Floating
“87.5”	String
‘4’	Character
4	Integer
4.0	Floating
“true”	String
false	Logical
“mehran”	String

Variables

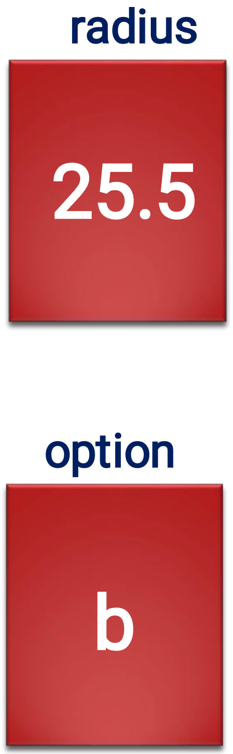
- In programming, a variable is the memory (RAM) location that can store the data temporary.
- Every program uses some sort of data and each data item is stored temporarily in the variables.
- Every variable has:
  - **Name**
  - **Data type**
  - **Value**

Variables

- The name of the variable is called as the identifier.

- Consider two of the examples.
- In first example, we have a variable whose name is radius,
its data types is floating and its value is 25.5.

- In second example, we have a variable whose name is
option, its data type is character and its value is b.



radius

25.5

option

b

Variables

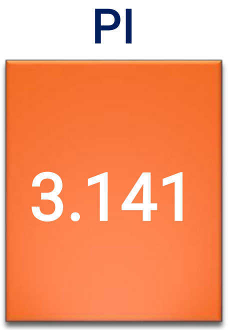
- The data type of the variable defines that what type of the data will be stored in a variable.
- Once the data type of variable is defined, it cannot hold any value of other data type.
- The value of the variable is changeable but its data type is not.

Constants

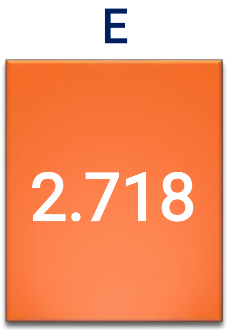
- In programming, a constant is a variable whose value remains fixed through out the program.
- In certain programs some sort of data remains unchangeable through out the program, hence we use constants to stored that data so that it can not be altered.

Constants

- Consider three of the examples.
- In first example, we have a constant whose name is PI, its data types is floating and its value is 3.141.

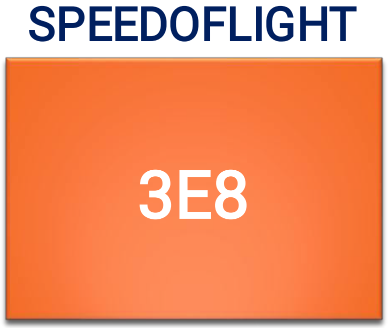


- In second example, we have a constant whose name is E (Euler’s number), its data type is floating and its value is 2.718.



Constants

- In third example, we have a constant whose name is SPEEDOFLIGHT, its data types is floating and its value is 3 x 10^8.



Rules for naming Variables and Constants

- The name of any variable for constant is called as **identifier**.
- There are certain rules to follow when setting the identifier.

- Rule 1: May contain letters (A-Z, a-z), digits (0-9) or an underscore sign.
- Rule 2: It must start with letter or underscore sign.
- Rule 3: Should not contain any space in between.
- Rule 4: Should not be like any keyword of language like, int, cout, float.
- Rule 5: It is case sensitive i.e. radius is not same as Radius or RADIUS.

Rules for naming Variables and Constants

- Rule 6: Must be unique i.e. no any two variables have same identifier.
- Rule 7: Must be relevant i.e. var1 is inappropriate identifier to store the area of circle, use area of area_circle instead.
- Rule 8: The identifier for constant is written with all capital letters. i.e. PI, SPEEDOFLIGHT, E etc.

Rules for naming Variables and C

Identifier	Valid/Invalid	Remarks
first_number	Valid	
first number	Invalid	Must not contain space in between
number_1	Valid	
1st_number	Invalid	Must start with a letter or underscore sign
first#	Invalid	Must not contain invalid character $
int	Invalid	Must not be like keyword
firstNumber	Valid	
int_first_number	Valid	
first_number_$	Invalid	Must not contain invalid character #
1st_#	Invalid	Must start with a letter or underscore sign Must not contain invalid character #

Do it yourself

Rules for naming Variables and C

Identifier	Valid/Invalid	Remarks
number of items		
#_of_items		
price/item		
number_items		
itemCount		
item#Price		
itemPrice_1		
5_itemPrice		
ITEM_PRICE		
$_per_item		

Operators

- Computer performs several operations on the data. The operations are represented by symbols known as operators.

- Operands are the data on which the operation is performed.

- Operand may be constant or stored in one of the variable.

Operators

- Example 1: a + b
- + is the operator
    - a and b are the operands
    - Both a and b are variables
- Example 1: 2 * length
    - * is the operator
    - 2 and length are the operands
    - 2 is constant and length is a variable

Arithmetic Operators

Operator	Symbol	Example Operation	Resultant
Addition	+	10 + 26	36
Subtraction	-	10 - 8	2
Multiplication	*	4 * 20	80
Division	/	25/5	5
Remainder/Modulus	%	13 % 5	3
Power	^	2^4	16

Relational Operators

Operator	Symbol	Example Operation	Resultant
Greater Than	>	5 > 6	false
Less Than	<	10 < 15	true
Greater Than or Equals To	>=	41 >= 90	true
Less Than or Equals To	<=	16 <=16	true
Equals To	==	15 == 19	false
Not Equals To	!=	14 != 80	true

Logical Operators

Operator	Symbol	Example Operation	Resultant
AND	&	true & true	true
		true & false	false
		false & true	false
		false & false	false
OR	¦	true ¦ true	true
		true ¦ false	true
		false ¦ true	true
		false ¦ false	false
NOT	!	! true	false
		! false	true

Logical Operators (AND)

A	B	A & B
True	True	True
True	False	False
False	True	False
False	False	False

Logical Operators (OR)

A	B	A ¦ B
True	True	True
True	False	True
False	True	True
False	False	False

Logical Operators (NOT)

A	! A
True	False
False	True