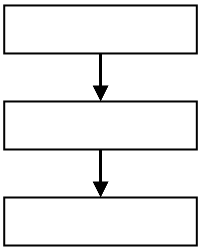
Abstract Data Type (ADT)
M.A. El-dosuky

[1] State the scientific name for each one:-

- disciplined approach to design, production, maintenance of programs **(software engineering)**
- A toolbox contains the computers and their peripheral devices **(hardware)**
- A toolbox contains operating systems, text editors and compilers **(software)**
- A toolbox contains the algorithms that we use to solve common programming problems, as well as data structures for modeling the information processed by our programs **(ideaware)**
- A step-by-step description of the solution to a problem **(algorithm)**

[2] Draw and explain the Software developing life Cycle

1. **<u>problem analysis</u>: understand the nature of the problem**
2. **<u>requirement definition</u>: what the program must do?**
3. **<u>implementation of the design</u>: coding the program in a computer language**
4. **<u>testing and verification</u>: detecting and fixing errors**
5. **<u>delivery</u>: turning over the program to the customer or user.**
6. **<u>operation</u>: using the program**
7. **<u>maintenance</u>: making chance to fix operational errors**



[3] Define the following terms

1. Abstraction,: **it is a model of a complex system that includes only the essential details. It is separation of a data type's logical properties from its implementation**
2. Abstract Data Type ADT : **A data type whose properties (domain and operations) are specified independently of any particular implementation**
3. Information Hiding: **The practice of hiding the details of a function on data structure with the goal of controlling access to the details of a module or structure**
4. Domain: **The set of all allowed values for a variable.**

   Domain of char is ASCII characters, of int is maxint..maxint
5. Data structure: **A collection of data elements whose organization is characterized by accessing operations that are used to store and retrieve the individual data elements; the implementation of the composite data members in an abstract data type**
6. Function Decomposition: **Top-down, the problem is first broken into several large parts. Each of these parts is, in turn, divided into sections, the sections are subdivided, and so on. The important feature is that details are deferred as long as possible as we move from a general to a specific solution.**

[4] Write a program to print the size of C++ data types.

```
#include <stdio.h>
int main( ){
        printf("\nA char  is %d bytes", sizeof(char) );
        printf("\nAn int  is %d bytes", sizeof(int) );
        printf("\nA short  is %d bytes", sizeof(short) );
        printf("\nA long  is %d bytes", sizeof(long) );
        printf("\nAn unsigned char  is %d bytes", sizeof(unsigned char) );
        printf("\nAn unsigned int  is %d bytes", sizeof(unsigned int) );
        printf("\nAn unsigned short  is %d bytes", sizeof(unsigned short) );
        printf("\nAn unsigned long  is %d bytes", sizeof(unsigned long) );
        printf("\nA float  is %d bytes", sizeof(float) );
        printf("\nA double  is %d bytes", sizeof(double) );
        return 0;}
```

```
Program output:
        A char  is  1 bytes
        An int  is  2 bytes
        A short  is 2 bytes
        A long  is  4 bytes
        An unsigned char  is  1 bytes
        An unsigned int  is  2 bytes
        An unsigned short  is  2 bytes
        An unsigned long  is  4 bytes
        A float  is  4 bytes
        A double  is 8 bytes
```

[5] (√×) Identify True or False and rewrite the correct version of False one:-

- Computer program is a list of instructions provide to computer to achieve specific task **(True)**
- Ideaware is a toolbox that contains programming methodology, Software concepts, and tools for measuring, evaluation and correctness of our programs**(True)**
- The programs we write are abstraction**(True)**
- An abstraction is a model of a complex system that includes only the essential details **(True)**
- Information hiding makes certain details inaccessible to programmer at higher levels**(True)**
- The concepts of abstraction and information hiding are fundamental principles of Hardware engineering **(False→ software)**
- Functional Decomposition One method for designing software **(True)**
- abstract data types are viewed from three different perspectives: their specification, their application, and their implementation. **(True)**
- *Data Abstraction:* separation of a data type's logical properties from its Implementation**(True)**

[6] Write program that simulate rolling of two Dice and print value of each one and the total sum.

- Using Data Structure ..Write the Abstract Data Type (ADT) for our Pair of Dice (Characteristics and Operations) and Using it to write the program

[7] Following program use Dice ADT, Identify errors and rewrite the correct program.

Dice ADT

Characteristics

The value of the first die, a random integer value from 1 to 6, name it (dice_1),
The value of the second die, a random integer value from 1 to 6 (dice_2),
The total sum of the two readings (sum)

Operations

int roll() randomize the values of the two dice and return the total sum
int die_1() return the reading of the first die
int die_2() return the reading of the second die

```
// Dice ADT
# include <iostream.h>
# include <stdlib.h>
int dice_1, dice_2, Sum;
// roll operation //
int roll( ){
        randomize();
        dice _1 = random(5) + 1;
        dice_2 = random(5) + 1;
        return dice _1 + dice_2;
        }
/* Dice1 operation */
int die_1 ( )
{
        return dice_1; }

/* Dice2 operation */
int die_2( )
{
        return dice_2; }

// test program to demonstrate the Dice ADT //
int main()
{
        cout << " first die : " << die_1 << endl;
        cout << " second die : " << die_2 << endl;
        cout << " Sum for roll is : " << roll() << endl;
        return 0;
}
```