Data Structures

By: Sayed Hassan Adelyar

Introduction to Data Structures

- Data Structures
  - Arrangement of data in computer memory
  - Algorithm efficiency depends on DS
  - Representation & Operations
  - Provide an important OOP goal: component reuse

Introduction to Data Structures

- Today's computers are data processors.
- The data is stored in RAM.
- The speed of CPU is increased continuously.
- The speed of search and update operations depend on the representation of data in computer memory.
- There are many different methods and techniques for storing data in computer memory. These methods and techniques are called data structures.

Introduction to Data Structures

- Each data structure has advantages and disadvantages.
- Storing data in such a way that brings efficiency, speed and ease of implementation is a fundamental issue in computer science. It is like a problem which needs continuous efforts for solution.

Introduction to Data Structures

- **Data structures** include:
- **Arrays**,
- S**tack**,
- **Queue**,
- **Linked lists**,
- **Binary search trees**,
- **Skip Lists**,
- H**ash tables,**
- **Heap**,
- **graph**, & others.
- **Algorithms manipulate** the **data** in these structures in various ways, such as **searching, sorting, and updating (inserting and deleting).**

Introduction to Data Structures

- In **today programming**, **object-oriented** is the **framework** for **building robust** and **re-useable** software.
- One of the **main ideas** of the **OO approach** is that data should be presented as being **encapsulated** with the **methods** that **access** and **modify** them. That is rather than simply viewing data as a collection of bytes and addresses, we think of data as **instances** of an abstract data type (**ADT**) that includes a **repertory** of **methods** for performing **operations** on the **data**.

Introduction to Data Structures

- **Class**
  - **Defines** the **structure** of an **object**. **Class definition** include **variable** and **methods**.
  - Java mechanism to create objects and methods
  - Classes have members which are:
    - Data
    - Methods
    - Constructors
- The separation of specification from implementation is an example of information hiding.

Programming Example for class (Rectangle)


```
class rectangle {
private int length;
private int width;
public rectangle(int len, int w) {
     length = len;
    width = w;
}
public int getlength() {
     return length;
}
public int getwidth()
{
     return width;
}
public int getarea()
{
 return width*length;
}
}
```

Introduction to Data Structures

```
class rectangleapp
{
public static void main(String [] args)
{
rectangle box = new rectangle(4,8);
System.out.println("Length is: "+ box.getlength());
System.out.println("Width is: "+ box.getwidth());
System.out.println("Area is: "+ box.getarea());
}
}
```

Introduction to Data Structures

- A **class** can has **two kind** of **variables**: **instance** variable and **class** variable.
- **Instance variable** values can be **different** for **each object**.
- **Class variable** use **static keyword** and they keep **only one copy** for **each object**.