بسم الله الرحمن الرحيم

Computer Programming
CSC-113

Intro

Engr. Muhammad Faisal
Senior Assistant Professor,
Department of Software Engineering
Bahria University, Karachi Campus,
Contact: mfaisal.bukc@bahria.edu.pk

Your Responsibilities

- As a Student
  - **Attend classes regularly**
  - Ask questions if you have any queries regarding course material or anything
  - Submit assignments on time
  - Don’t miss quizzes, assignments, mid term and final examinations – Can get good result ☺
  - **No Plagiarism is allowed in any sort of a written material – Write in your own words**

Course Policy

- Assignments
  - Assignments are due at the beginning of class
  - Late assignments will not be accepted
  - All works have to be done independently except in case of group assignments
  - Students handing in similar assignments will receive a grade of 0 (ZERO) and face possible disciplinary actions
- Makeup Exams
  - In accordance with university regulations, i.e. students should bring a valid excuse authenticated through valid channels immediately within first week after the exam
  - Otherwise they will be considered absent and no makeup exam for them
- Mobile Phones
  - Cell phones are restricted during class. Cell phones must be turned off during the lecture. If your cell phone rings during the class, you may be asked to leave

Course Policy

- Attendance
  - Students are expected to attend all classes
  - If a student misses **12 hours in total** without an acceptable reason, the student will be barred from the final exams
- Code of Conduct
  - The assignments, quizzes and exams need to be done individually. Copying of another student's work or code, even if changes are subsequently made, is inappropriate, and such work or code **will not be accepted**
  - The University has very clear guidelines for academic misconduct, and they will be enforced in this class
  - However, in case of group assignments, copying of another group’s work or code, even if changes are subsequently made, is inappropriate, and such work or code **will not be accepted**

Course Description

- In this course the student will gain a broad understanding of modern computer programming. The student will acquire introductory skills in problem analysis, solution design, and program construction. Through practical programming activities, the student will gain an appreciation of the nature of computer programming.

Course Learning Outcomes

CLO	Statement	Bloom's Taxonomy	Associated PLO
1	Describe basic concepts related to computer, programming concepts and constructs of a structured programming language including lexemes, expressions, statements and methods	C1	PLO-1
2	Analyze programs with respect to their performance, correctness, possible output, errors and exception.	C4	PLO-2
3	Apply knowledge and understanding to solve computing problems and project their solutions under specific requirements	C3	PLO-3
4	Present an advanced computer programming topic confidently and professionally.	A2	PLO-9

Marks Distribution

Evaluation Instruments (EI)	Marks
Quizzes	10
Assignments	20
Mid Term Examination	20
Final Examination	50
Total:	100

Mapping of CLOs to Course Evaluation Instruments (EI)

EI	CLO’s			
	CLO 1	CLO 2	CLO 3	CLO 4
Assignments			x	x
Quizzes	x	x	x	
Midterm Exam	x	x	x	
Final Exam	x	x	x	

Course Resources

Text Book:

- Problem Solving and Programming Concepts, Maureen Sprankle, Jim Hubbard, , 9th Edition, Pearson.

Reference Books:

1. Fundamentals of Computer Programming with CSharp, Nakov
2. Visual C# How to Program, Harvey M. Deitel & Paul J. Dietel., 6E.
3. Professional Visual Studio 2019, Bruce Johnson

Online References:

- https://www.tutorialspoint.com/csharp/
- https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/
- https://mva.microsoft.com/en-US/training-courses/c-fundamentals-for-absolute-beginners-16169

Course Contents

- ➢Define a computer system and its parts
- ➢New technologies in computer
- ➢Describe the evolution of computer languages
- ➢Explain algorithm, flowcharts, pseudocodes
- ➢Define operators, their types and usage
- ➢Conditional Statements [if, switch]
- ➢Iterative Statements [Looping Constructs]
- ➢Define Arrays, its types, different search and sorting techniques
- ➢Define Methods and its usage
- ➢Structure and its usage
- ➢Pointers and its usage
- ➢File Handling

Introduction to Computers

Session 1

Objectives

- ❑Define a computer system
- ❑Describe parts of a computer
- ❑Explain the computer system startup process
- ❑List new technologies in computer

Introduction

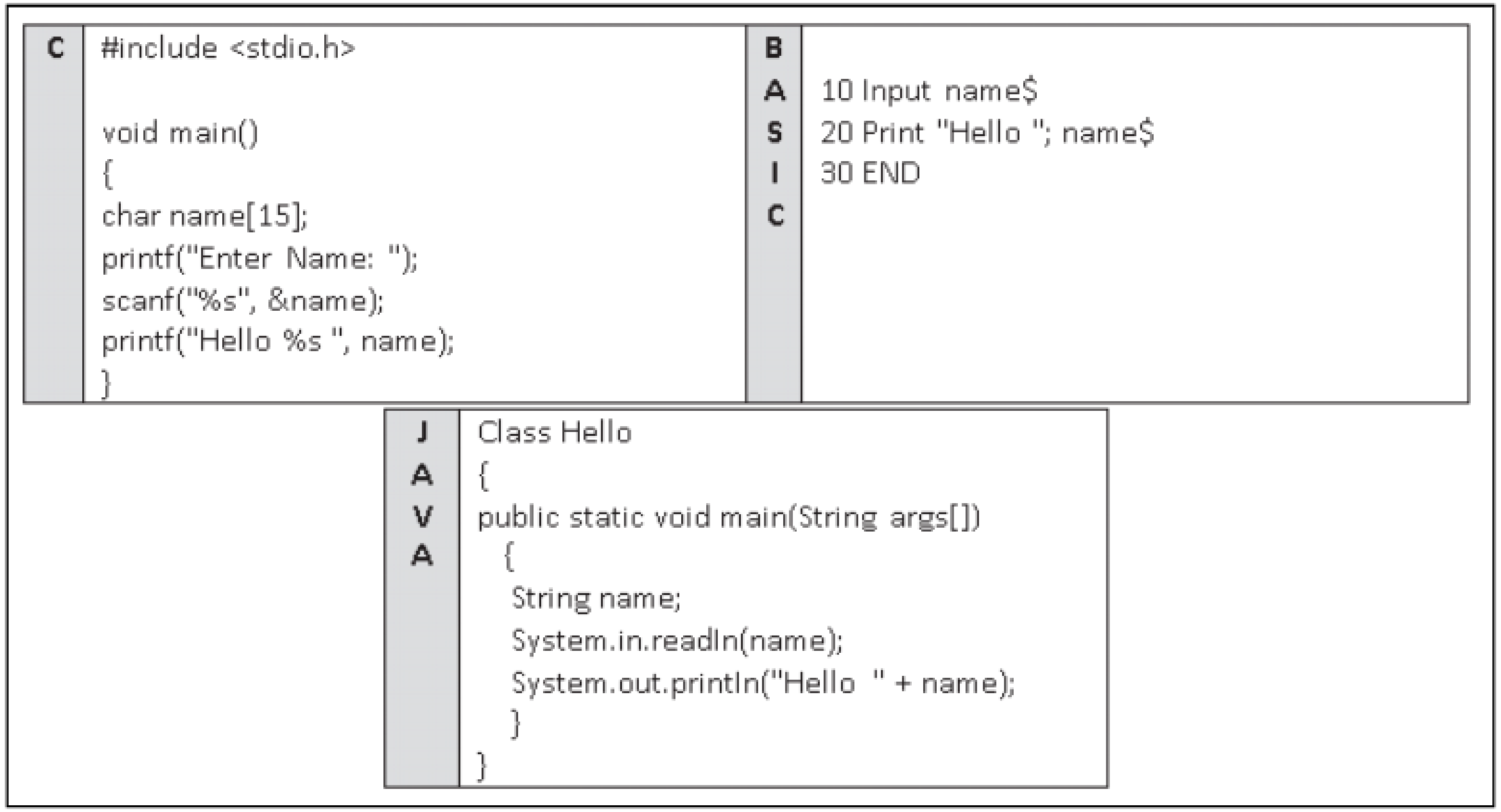
- Computers can perform different functions and calculations, but a computer cannot take a decision as a human being.

- Computers have to be instructed or programmed to perform certain task.

- For this purpose, several programming languages have been developed to help programmers write code to meet user’s requirement.

Introduction

- The basic approach to write the programming code is almost similar for all programming tools.



C

```
#include <stdio.h>

void main()
{
char name[15];
printf("Enter Name: ");
scanf("%s", &name);
printf("Hello %s ", name);
}
```

BASIC

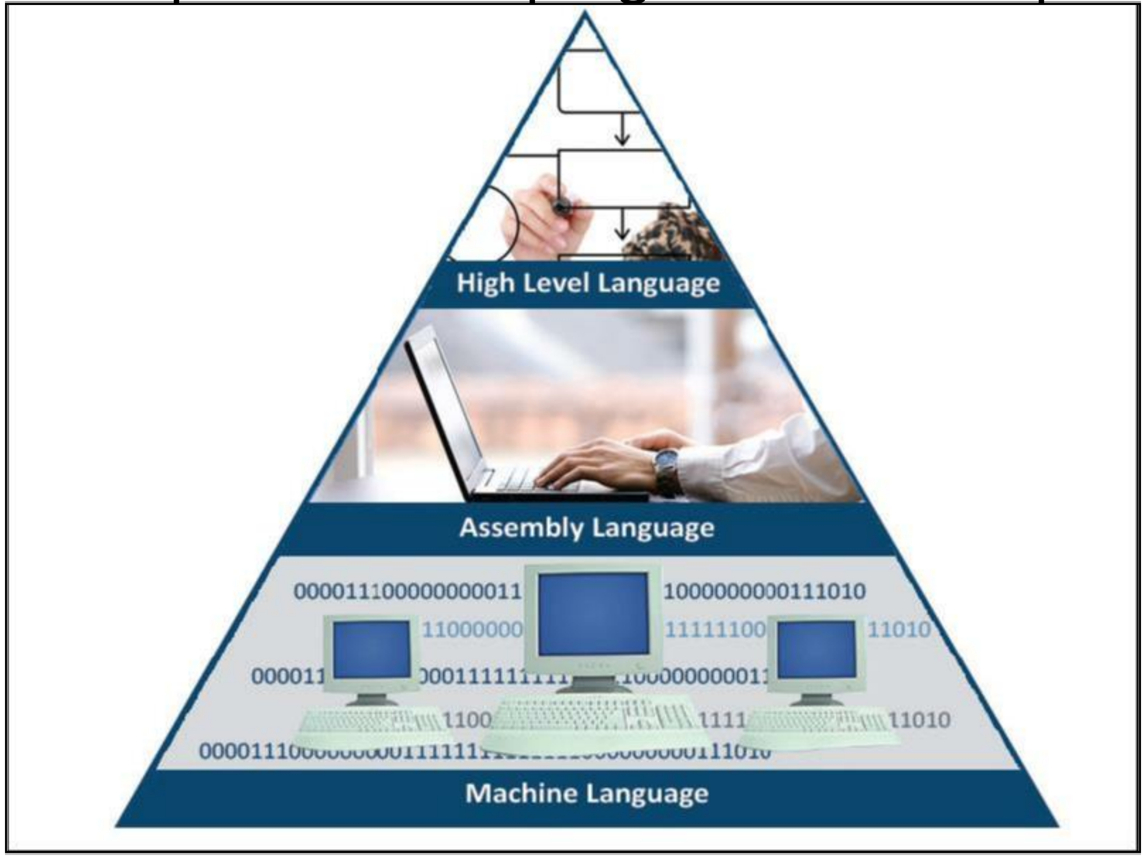
```
10 Input name$
20 Print "Hello "; name$
30 END
```

JAVA

```
Class Hello
{
public static void main(String args[])
  {
   String name;
   System.in.readln(name);
   System.out.println("Hello  " + name);
   }
}
```

Computer Languages

- Developed for the purpose of communication between the different components and programs of a computer.



Computer Languages

❑Machine Language

The first **generation languages**, or **1GL**, are low-level languages that are machine language

❑Assembly Language

The **second-generation languages**, or **2GL**, are also low-level assembly languages. They are sometimes used in kernels and hardware drivers

❑High Level Language

A **high-level language** (HLL) is a **programming language** enables a **programmer** to write programs that are more or less independent of a particular type of **computer**. They are closer to human **languages.**

➢Third Generation Languages

The third-generation languages, or **3GL**, are high-level languages, such as C, C++, Java, JavaScript, and Visual Basic.

➢Fourth Generation Languages

The **fourth-generation languages**, or **4GL**, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts examples include Perl, PHP, Python, Ruby, and SQL.

➢Fifth Generation Languages

The **fifth-generation languages**, or **5GL**, are programming languages that contain visual tools to help develop a program. Examples of fifth generation languages include Mercury, OPS5, and Prolog.

Introduction

- A computer can be defined as a system that responds to a set of instructions in a well-defined manner
- The computer system consists of different parts namely:
  - Central Processing Unit (CPU)
  - Motherboard
  - Random Access Memory (RAM)
  - Hard disk
  - Main cabinet with all the peripheral devices connected to the motherboard
  - Operating System (OS)

Various Components of Computer System

❑ The different mandatory components of a computer system are as follows:

Processor

- The main component of a computer that executes all processes and instructions supplied by memory unit

RAM

- This area in the computer stores all the instructions (processes) and information of system. It is a volatile memory

ROM

- ROM is a memory chip, which contains in-built programs written into it by manufacturers. It is a non-volatile memory

HDD

- It is a non-volatile memory because it retains information, unless the information itself is deleted or overwritten. It can store large amount of data

Monitor

- The most commonly used output device is the monitor. It is used to display a variety of information

Keyboard

- The keyboard is one of the most common input devices. Keyboard convert numbers, letters, and other special characters into digital signals, which the system understands

Mouse

- A mouse is an input device that controls a pointer, which is displayed on the monitor. The mouse is used to click and drag objects on the graphical interface as well as to select or activate options

Types of Computers

- ❑ Computers can be classified based on their size and processing power

- ❑ The different types of computers are as follows:

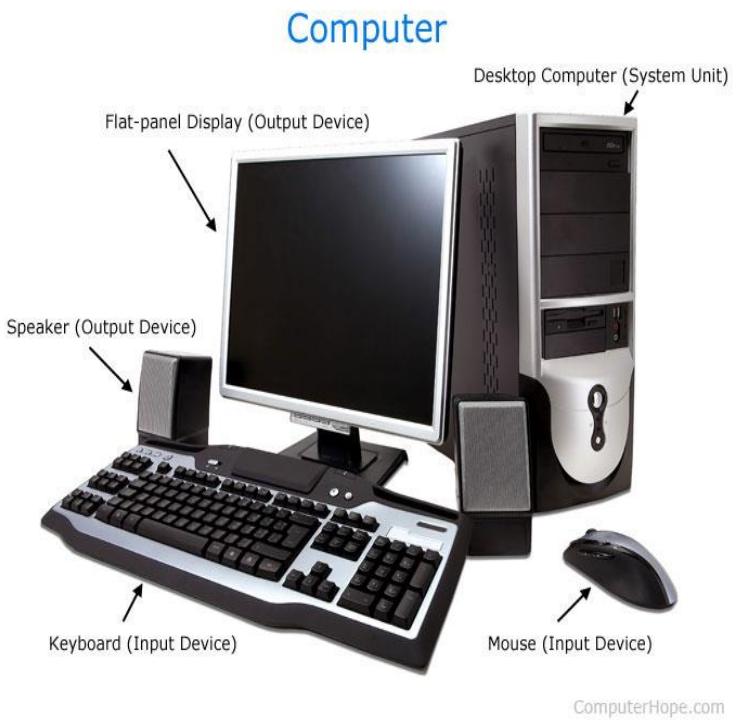
  - Personal Computers
  - WorkStation
  - MiniComputer
  - SuperComputer
  - Mainframe Computer

Personal Computers or Microcomputers 1-5

- They are relatively small as compared to supercomputers and mainframes
- They are inexpensive and designed for individual users
- They are used in business, homes, and schools for surfing Internet, playing games, listening to music, and so on
- They come in different forms such as desktop computers, laptop computers, tablet PCs, Personal Digital Assistants (PDAs), and so on

Personal Computers or Microcomputers 2-3

- Desktop Computers
  - These computers are not portable and are designed to fit on desk
  - They are much smaller in size, easy to use, and cheap as compared to other types of PCs
  - They are most commonly used in offices, homes, cyber cafe, and so on



Laptop Computers

- Laptop computers are light-weight and portable

- The physical size of a laptop is the primary factor that determines the cost of a laptop

- Laptops that are smaller in size are known as notebooks

Personal Computers or Microcomputers 3-3

- ❑ Tablet PCs
  - ➢ Tablet PCs are similar to notebook PC and have a touch screen or pen enabled interface
  - ➢ They are very useful and popular for writing notes in the fields of law, education, and medicine
  - ➢ This type of computer offers mobility for a user who does not have enough space to work with a desktop, laptop, or notebook PC

- Handheld Computers or PDA
  - A PDA also known as a palmtop computer or a pocket PC, is a handheld miniature computer
  - Mobile phones that are integrated with PDA functionality are called Smartphones
  - They are smaller and lighter in weight and can be carried in a pocket
  - These devices hold data such as schedules, notes, appointments, address book, and many more

Workstation

A workstation can be defined as a type of computer that is mainly used for technical and scientific applications, desktop publishing, and so on

They are used by a single person and are connected to a local area network (LAN)

It has a moderate computing power and a high quality graphic capability

Workstations have built-in network support, mass storage device, graphical user interface, and so on

MiniComputer

- Minicomputer is a multi-user computer and supports hundreds of users concurrently
- They are smaller to mainframe computers as far as speed, performance, and storage capacity are concerned
- When compared with mainframes, they are less expensive

SuperComputer

SuperComputers are fast computers that can execute trillions of instructions per second

They are used by applications that require intensive numerical computations such as weather forecasting, nuclear energy research, oil and gas exploration, and so on

Mainframe is a large computer that can support hundreds or thousands of users simultaneously

Computer Software and its Types 1-3

Software is a generic term for organized collections of computer data and instructions

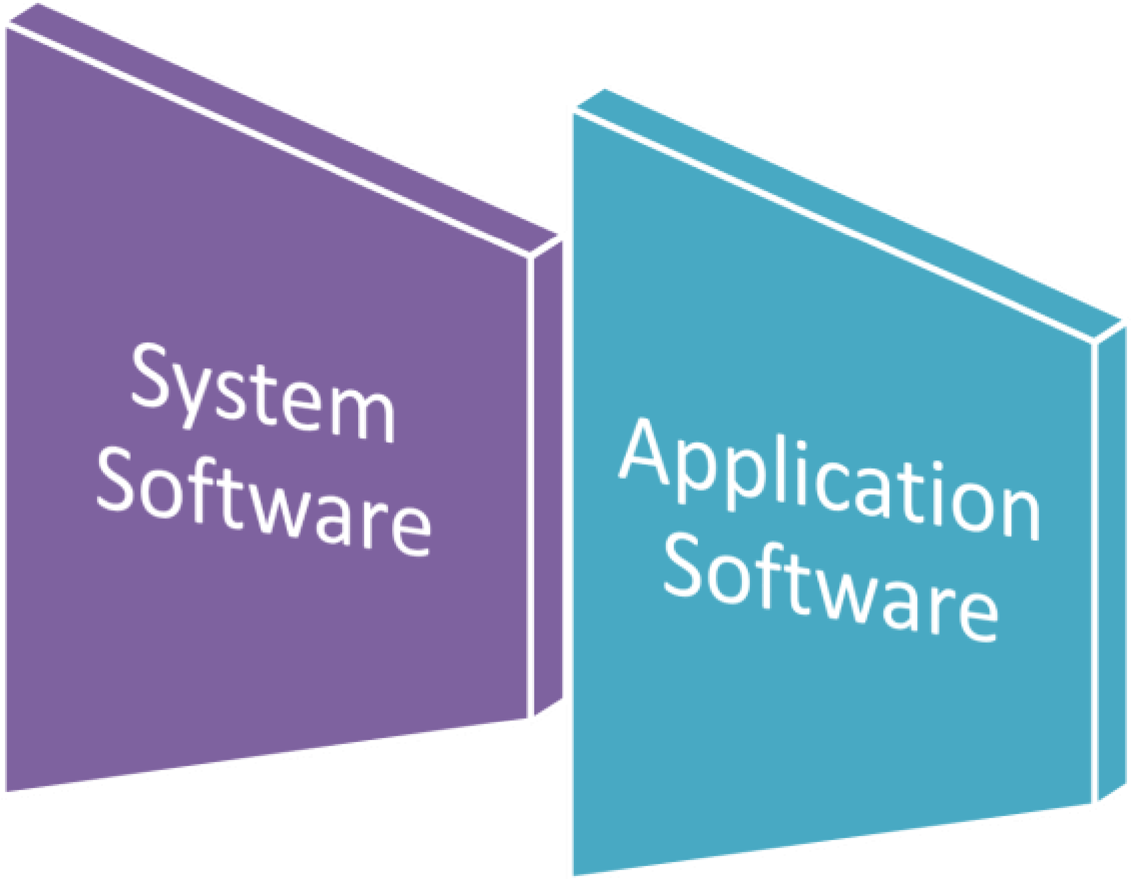
A sequence of instructions is specified to a computer to solve a problem

These sequence of instructions are written in a language, which can be comprehended by a computer and they form a computer program

The program controls the processing of the computer, which performs specifically as the program instructs

Computer Software and its Types 2-3

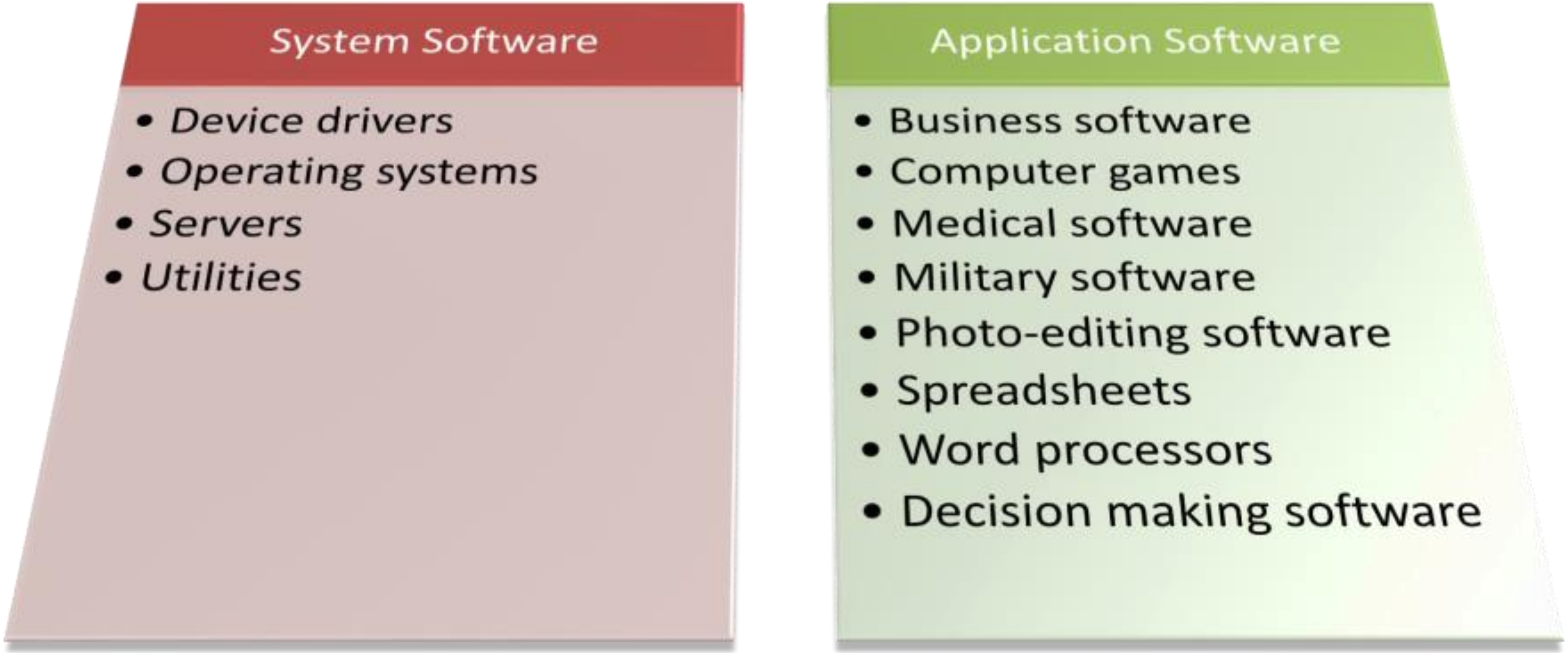
- There are two types of software, namely:



System Software

Application Software

Computer Software and its Types 3-3



System Software

- *Device drivers*
- *Operating systems*
- *Servers*
- *Utilities*

Application Software

- Business software
- Computer games
- Medical software
- Military software
- Photo-editing software
- Spreadsheets
- Word processors
- Decision making software

System Software

- Can be defined as a collection of programs which assist with program development and the execution of program
- **Types of systems software:**
  - Language translators
  - Loaders
  - Linkers

Language translators

Three core objectives of language translators are:

- Instruction statement conversion
- Source code error detection
- Object code optimization

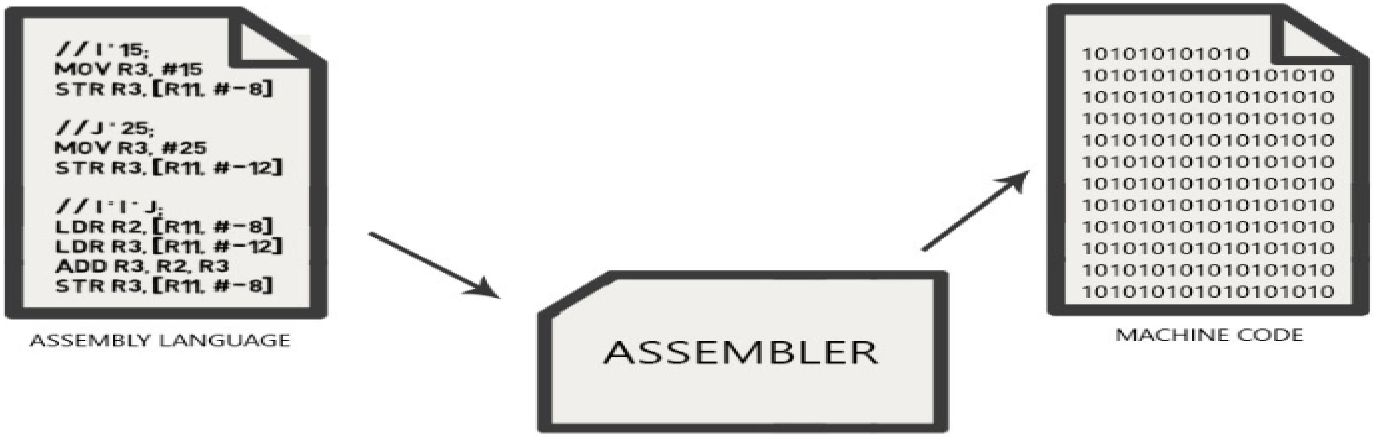
Types of translators

The different types of translators are

- Assembler
- Compiler
- Interpreters

Assembler

- Translates assembly language programs into machine language



```
// I · 15;
MOV R3, #15
STR R3, [R11, #-8]

// J · 25;
MOV R3, #25
STR R3, [R11, #-12]

// I · I · J;
LDR R2, [R11, #-8]
LDR R3, [R11, #-12]
ADD R3, R2, R3
STR R3, [R11, #-8]
```

```
101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
1010101010101010
```

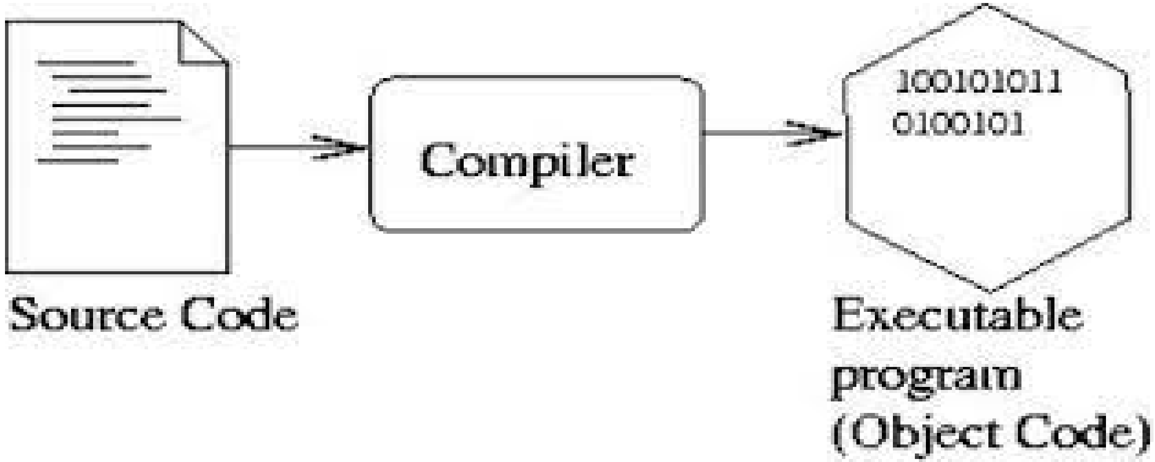
www.educba.com

Compilers

- Translates a source program in a high level language into its equivalent object program in a machine language.
- The object program may then be executed.
- The translation process is applied to the source program as a whole.
- There is not always a direct correspondence between a source code and one or more object program instructions

Cont….

- Translate the entire source code into its equivalent object code.
- The object code is then loaded into the main memory and executed.
- Object code is always produced.
- The program needs be translated only once.
- Short execution time.
- More sophisticated
- Optimization can be incorporated



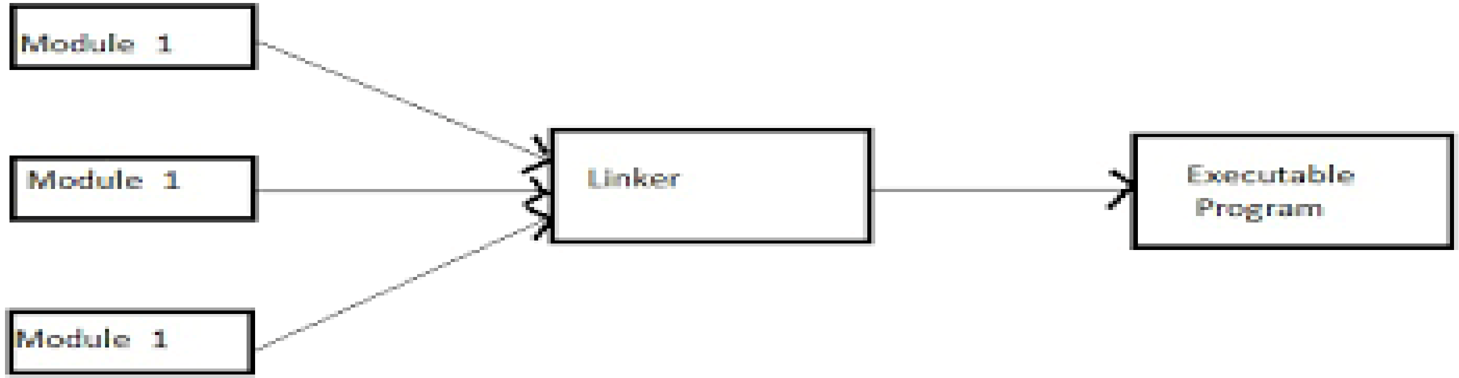
Interpreters

- Deal with the source code one instruction at a time, translating and executing each instruction before it goes on to the next.
- Object code is seldom produced, and even it is produced, is not retained.
- The program will be translated each time it is executed.
- Longer execution time.
- Simpler and easier to use.
- Optimization is virtually impossible

Linker

The program which are converted into machine language by the translators cannot be executed directly for the following reasons

- Program may call subroutines that were not converted to machine language at the same time as the current program and so their addresses are unknown
- Program may be a subprogram and requires a calling program to work
- Program may not be placed in memory at the starting address assumed by the translator

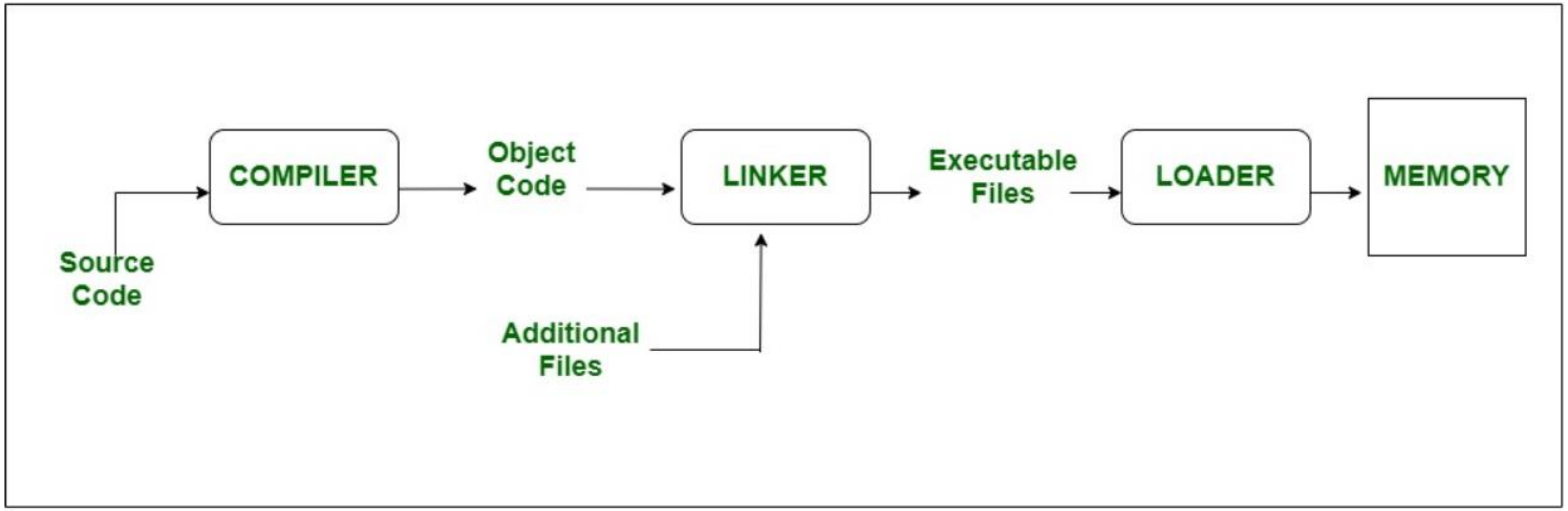


Cont…..

- A linker or linkage editor converts the object code in a load module that can be placed in the memory and execute.
- It combines separate module of machine code before execution and after compilation. It enables a program to be linked to various utility programs. This is done by planting the appropriate addresses in all the call and return instructions, so that all the module are linked together properly.

Loader

- Loader is a program which accepts the object code, prepares these programs for execution and initiates the execution



Cont….

Loader must perform four functions:

- **Allocation:**

  Allocates space in the memory for the programs.

- **Linking:**

  Resolve symbolic references between object codes.

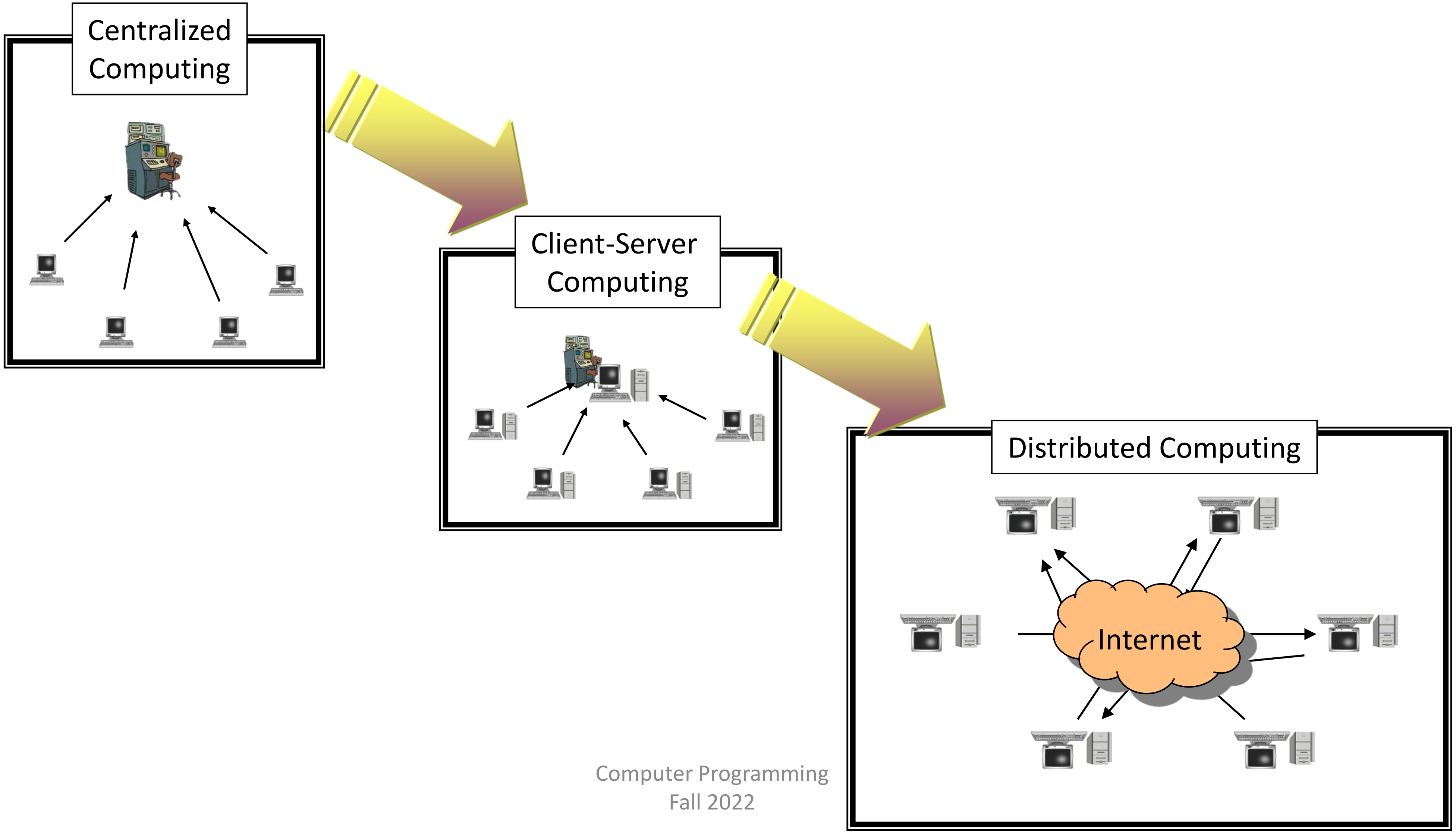
- **Relocation:**

  Adjust all address dependent locations to correspond to the allocated space

- **Loading:**

  Physically place the machine instruction and data into memory

Transformation in Computing



Cloud Computing

- Cloud computing evolved from a concept called virtualization
- Using virtualization, you can host multiple Operating Systems at the same time on a single machine
- A traditional application server may have just 5-10% utilization, whereas virtualized servers can reach 50-80% utilization
- By hosting more virtualized instances on fewer physical servers, you can lower costs for hardware acquisition, maintenance, energy, and cooling system usage

Grid Computing

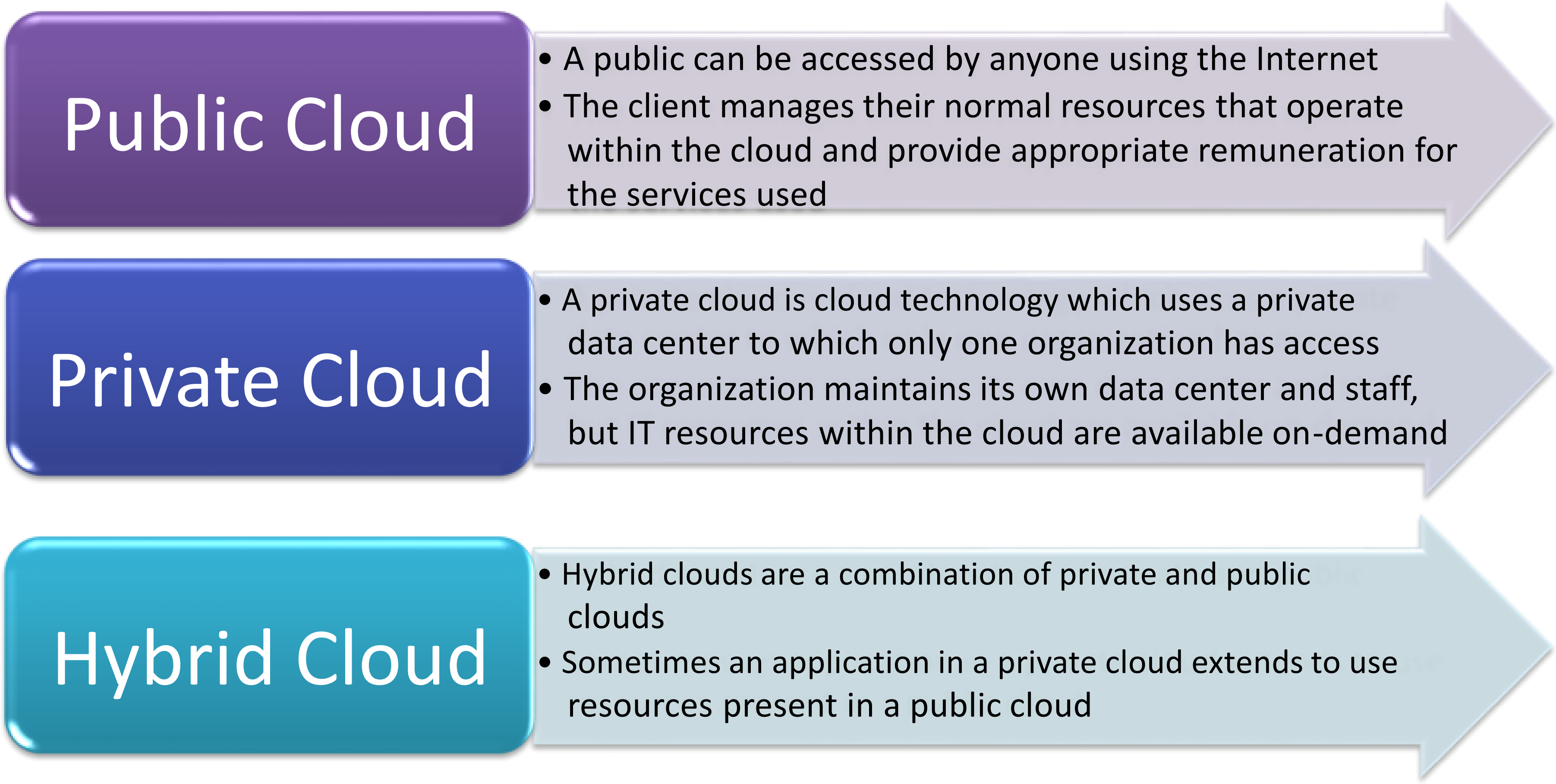
- Grid computing is defined as an interconnected computer system where the machines utilize the same resources collectively for solving a problem or reaching a common goal
- Grid computing usually consists of one main computer that distributes information and tasks to a group of network computers to accomplish a common goal

Utility Computing

- In utility computing, the computing resources and infrastructure is provided to the customer by the service provider
- The customer is charged according to the service usage

Cloud Computing 1-2

- Cloud computing is divided into three major cloud structures:



Public Cloud

- A public can be accessed by anyone using the Internet
- The client manages their normal resources that operate within the cloud and provide appropriate remuneration for the services used

Private Cloud

- A private cloud is cloud technology which uses a private data center to which only one organization has access
- The organization maintains its own data center and staff, but IT resources within the cloud are available on-demand

Hybrid Cloud

- Hybrid clouds are a combination of private and public clouds
- Sometimes an application in a private cloud extends to use resources present in a public cloud

Cloud Computing 2-2

- The different types of services provided in cloud computing platform are:
  - **Infrastructure as a service (IaaS)** - In this model, computers and other resources are provided to the users
  - **Platform as a service (PaaS)** - In this model, OS, program execution environment, database, and Web server are provided as a service
  - **Software as a service (SaaS)** - In this model, cloud providers install and operate the application software in the cloud and the users access it
  - **Storage as a service (STaaS)** - In this model, large service providers rents out storage infrastructures
  - **Security as a service (SECaaS)** - In this model, large service providers integrates into the corporate infrastructure's security services
  - **Data as a service (DaaS)** - In this model, data is provided to the user as and when required irrespective of geographic location or organization separation between the required the consumer
  - **Test Environment as a service** (TEaaS) - In this model, software and its data are hosted to be accessed by users using a Web browser over the Internet
  - **Desktop as a service** (DaaS) - In this model, desktop is virtualized
  - **API as a service** (APIaaS) - It enables the creation and hosting of APIs

Grid Computing 1-2

❑ Advantages of grid computing are:

- Large six figure Symmetric Multiprocessing (SMP) servers for application processing are not required
- Idle resources can be utilized much more efficiently by distributing jobs to idle servers or idle desktops
- A grid environment has modular structure and does not have single points of failure
- Systems can be upgraded without scheduling downtime
- Jobs are executed in parallel speeding performance

Grid Computing 2-2

- Disadvantages of grid computing are:

Large SMP are still used when applications that require high memory do not take advantage of MPI [Message Passing Interface]

A fast connection in between the computer resources is required

Some applications are required to be fine-tuned to take full advantage of the new model

Utility Computing 1-2

❑ Advantages of utility computing are:

The client is not required to buy all the hardware, software and licenses, instead, the client depends on the utility computing company to provide these services

It gives companies the option to subscribe to a single service and use the same software suite for the entire client organization.

Utility Computing 2-2

- ❑ Disadvantages of utility computing are:

If a utility computing company has financial difficulty or has frequent equipment problems, clients could get discontinued from the services even after paying.

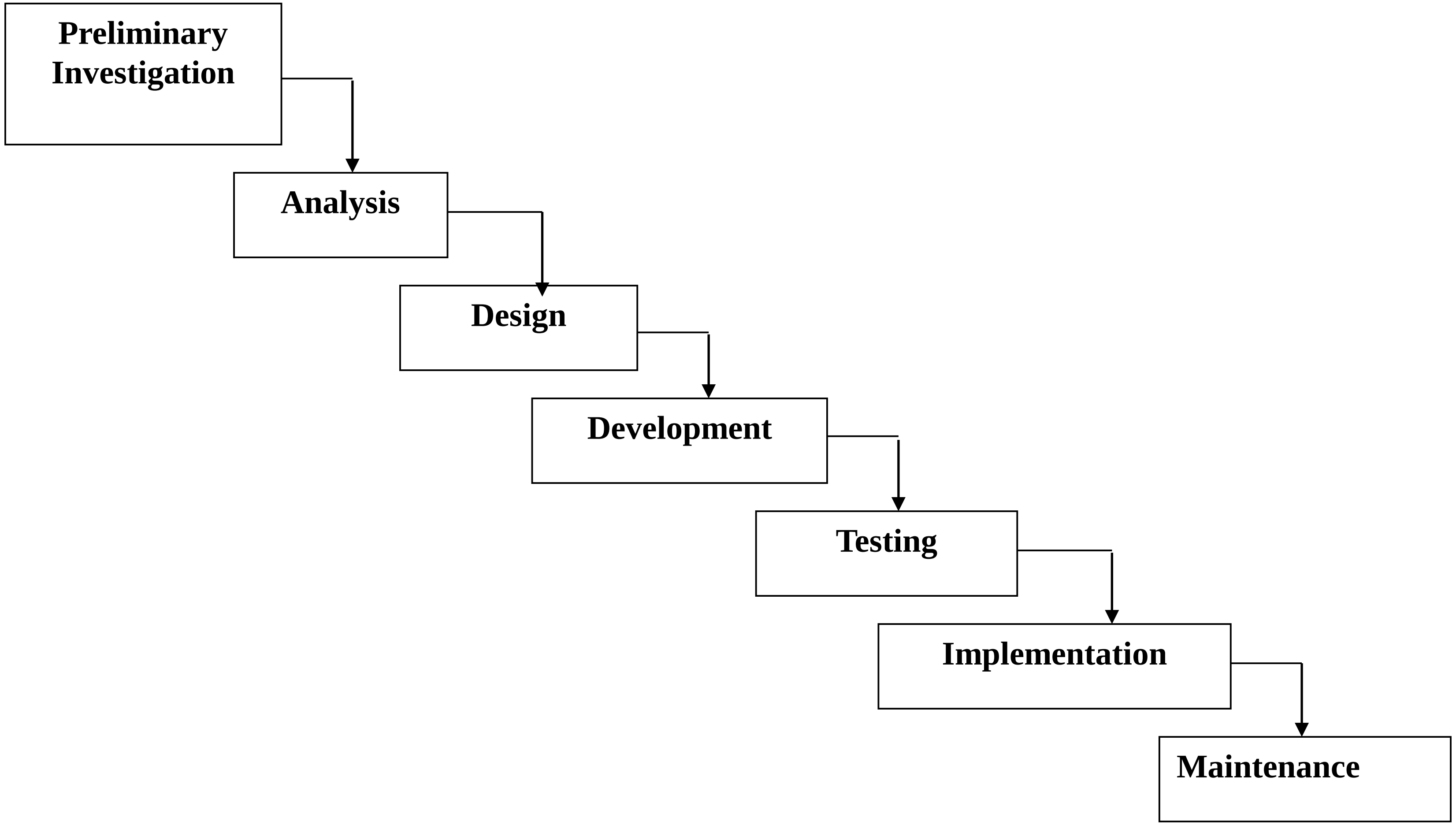
Utility computing systems are always targets for hackers. A hacker can access services without paying and sneaking around and exploring client files.

The Systems Development Life Cycle for Programming

The SDLC for programming follows the same phases as the SDLC for information systems development:

- Preliminary Investigation (Feasibility Study)
- Requirement Analysis (Analysis)
- Design of the System
- Software Construction
- System Testing
- System Implementation
- System Maintenance

Schematic view of phases in SDLC


Preliminary Investigation
Analysis
Design
Development
Testing
Implementation
Maintenance