Software Design and Analysis
CS-3004

Dr. Javaria Imtiaz,
Mr. Basharat Hussain,
Mr. Majid Hussain

Outline

- Introduction to the Course
- Course Objectives
- Why this course is important for you?
- Course Contents and Structure
- Course Project

About Dr. Javaria Imtiaz

- Assistant Professor
- PhD(SE) – FAST-NUCES (2021)
  - Full Funded Scholarship
- Research Associate – Quest UAV Dependability Lab (2018-current)
- Professional Trainer of SCM, Web Test Automation, DevOPs
- Over 9 years of Industry and Academia experience
  - Expertise: Web Test automation, Software Testing , Web application product lines, Java (SE & EE), UML, Model-driven Software Engineering, DevOPs.
- MS(SE) – FAST-NUCES (2015-2017)
  - Full Funded Scholarship
  - Distinction
- Instructor – FAST-NUCES (2014-2018)
- BS(SE) – IIUI (2010-2014)

About Mr. Basharat Hussain

- Lecturer
- PhD(CS) – COMSATS (In-progress)

- Over 24 years of Industry and Academia experience
  - Expertise: .NET, Programming, Design and Architecture, Software Testing , product lines, UML, Model-driven Software Engineering, DevOPs

- MS(CS) – IIUI (2014-2016)
  - Distinction

- Instructor – COMSTAS, RIPHAH, FAST-NUCES

About Mr. Majid Hussain

Contact Details

- **Dr. Syeda Javaria Imtiaz**
  - Email: javaria.imtiaz@nu.edu.pk
  - Office: Room (205-B, 2nd Floor Computer Science Block C)
- **Mr. Basharat Hussain**
  - Email: basharat.hussain@nu.edu.pk
  - Office: Room (209-D, 2nd Floor Computer Science Block A)
- **Mr. Majid Hussain**
  - Email: majid.hussain@nu.edu.pk
  - Office: Room (304-C, 3nd Floor Computer Science Block C)

- **Miscellaneous**
  - Office Hours: Will be Displayed outside my office.
  - 2 Classes of 1.5 hours

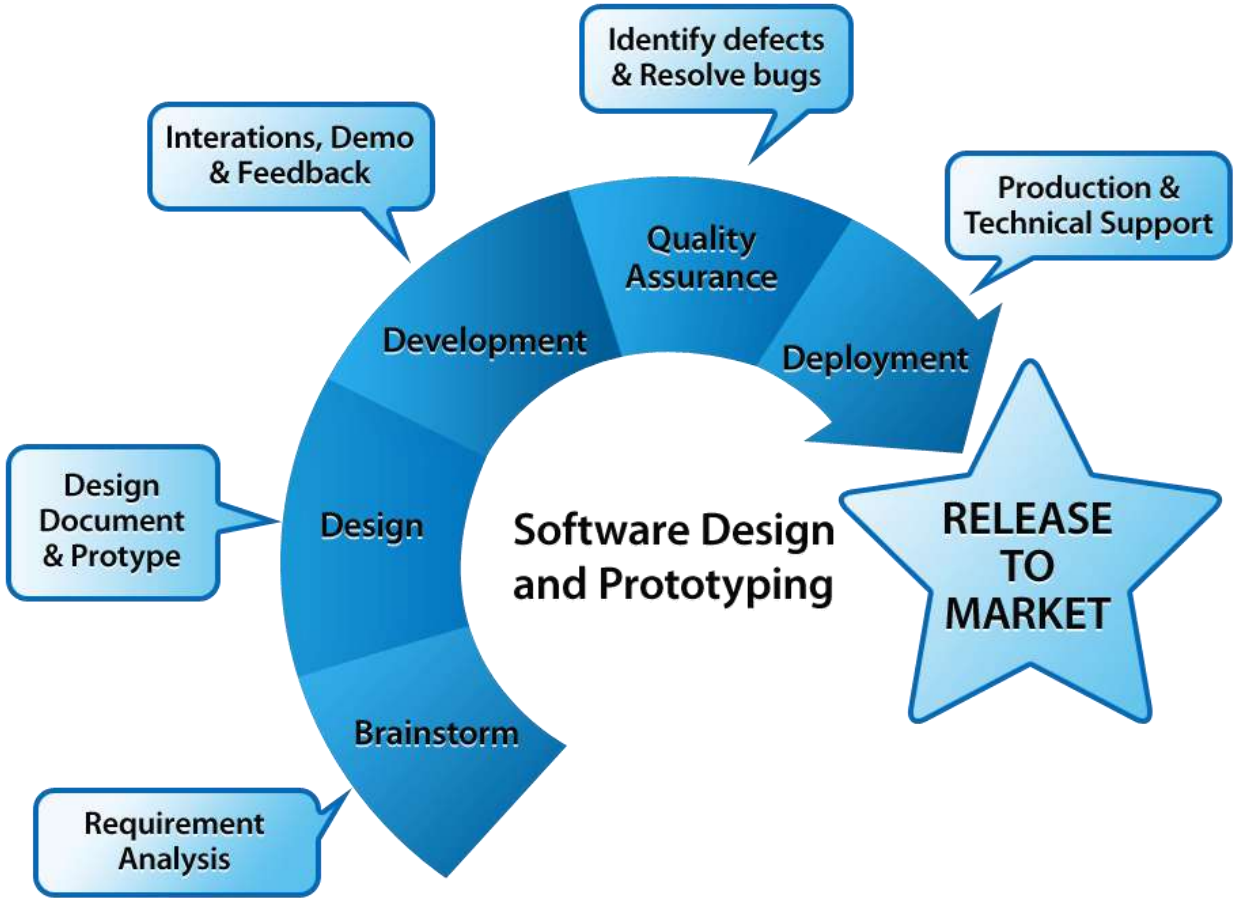
Lecture Format

- Start with Q & A
- Main Lecture
  - No short breaks in the middle
- Practice sessions on various case studies
- Designing/Modelling of software artifacts
- Translation of design artifacts into code

Why this course is important for you?

- Almost every large and complex industrial project follow a detailed design.
  - Every safety critical system must be well designed according to their safety standards.
- Industry hire requirement analysts and software designers to model the ongoing projects.
- Design artifacts carry a major weightage in FYP.

Software Development Process

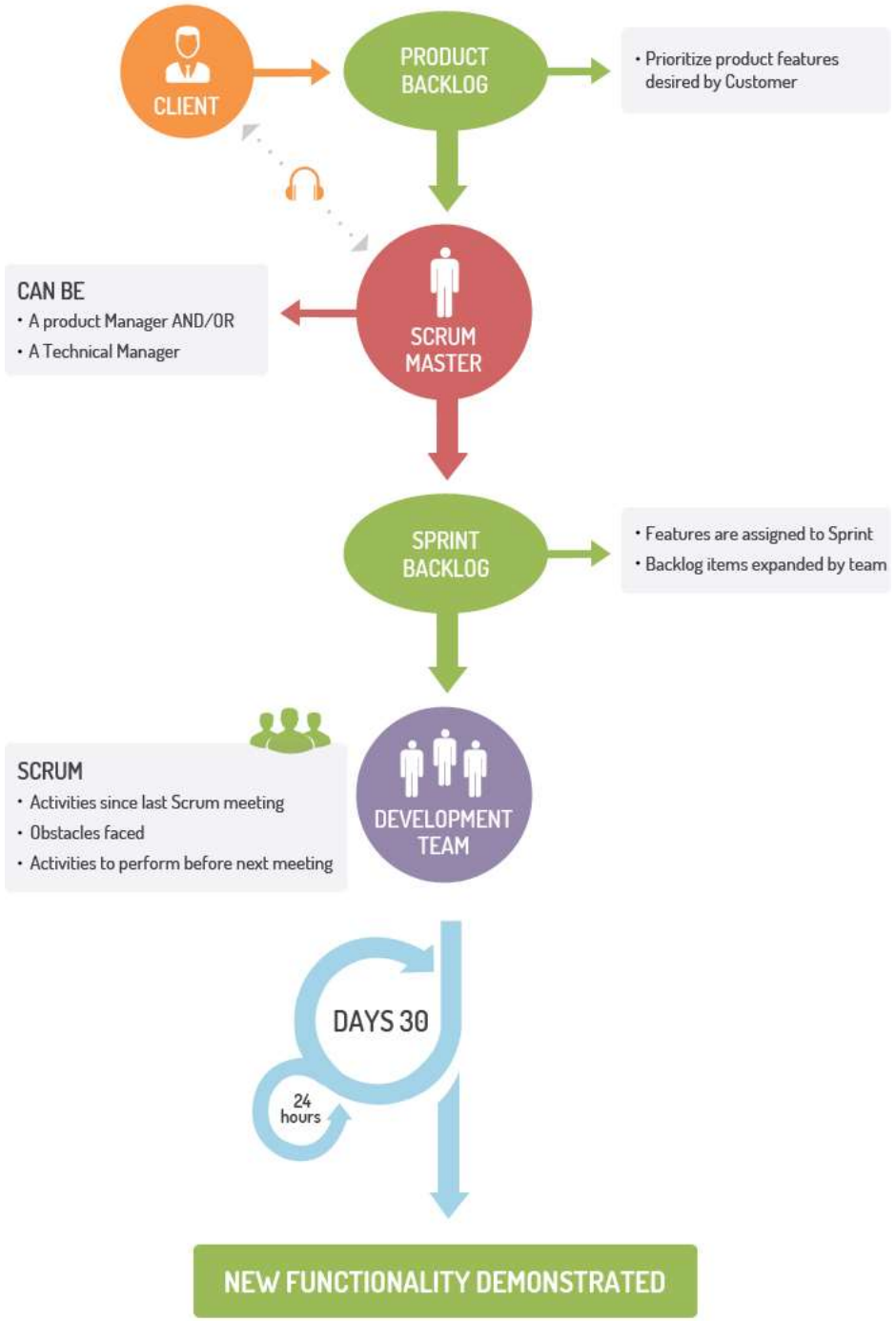


Software development Methodologies

- frequent alteration in the development project
- minimize risk by developing software in short time boxes
- changing requirements of the clients.

Cont..

- Follow continuous integration and continuous delivery model
- development and operational teams to perform everything simultaneously in development, quality assurance, security, and other operations.
- It makes the process faster and easier for businesses to process on time



Cont..

- For companies where the requirements are highly emerging and rapid changes are easily adhered to.
- daily meeting easily helps the developer to make it possible to measure individual productivity.
- Due to short sprints and constant feedback, it becomes easier to cope with the changes

Introduction to the Course

- A major engineering problem today
  - Predictable development of reliable software systems.

- The object-oriented paradigm
  - solutions to many development problems
  - provides a reliable design, complexity control, and reusability
  - Object-oriented methods can be divided into analysis and design, implementation languages, and distribution
  - Basic to the entire enterprise, however, are analysis and design, which has remained remarkably stable for years.

- Object-Oriented Analysis and Design
  - overall goals of the object paradigm, the selection of classes, the relationships among them, and their realization to implement systems

Introduction to the Course

- Object-Oriented Analysis and Design

  - Concepts and techniques necessary to effectively use system requirements to drive the development of a robust OO design model

  - Unified Modeling Language (OOAD/UML) 2.2

  - Focus on training by applying UML 2.2 notation to fundamental OOAD concepts including architecture, objects, classes, components, subsystems, stereotypes, relationships, and supporting diagrams.

  - Implementation

Traditional Software Development Flow

- When a software developer begins work with a new project, it's natural for him to choose to commence scripting the code immediately.
- Regardless of his software knowledge, if someone produces software without laying out a strategy for it, he might as well have been laying bricks without establishing a solid foundation.
- The building will fall no matter how robust the construction process and resource consumption are; similarly, the software would fail without a fundamental layout.
- Nobody wants their efforts to be for waste because they didn't have a plan for them, right? That's where software design comes in, a mechanism that simplifies important software operations in a certain way.

What is Software Analysis?

- **Analysis** emphasizes an investigation of the problem and requirements, rather than a solution.
  - For example, if a new online trading system is desired, how will it be used? What are its functions?

- "**Analysis**" is a broad term, best qualified, as in requirements analysis (an investigation of the requirements) or object-oriented analysis (an investigation of the domain objects).

What is Software Design?

- **Design** emphasizes a conceptual solution (in software and hardware) that fulfills the requirements, rather than its implementation.
  - For example, a description of a database schema and software objects.

- **Design** ideas often exclude low-level or "obvious" details—obvious to the intended consumers.

- **Design** can be implemented, and the implementation (such as code) expresses the true and complete realized design.

Useful analysis and design have been summarized in the phrase **do the right thing** (analysis), and **do the thing right** (design).

Course Objectives

- At the end of the course students will be able to:
  - Analyze problems from an object-oriented perspective
  - Apply an iterative, use case-driven, architecture-centric process to the development of a robust design model
  - Use UML 2.2 to represent the design model that take advantage of OO concepts of such as abstraction, encapsulation, inheritance, and polymorphism
  - Create OO designs and implement in Java
  - Apply design considerations through the use of patterns
  - Implement a pilot iterative OO project that leverages all key concepts learned in the course

Course Structure

- 3 Credit Hour Course (More applied than you think)
  - No labs
- Programming Language and Tools
  - Java
  - UML Case tools like Enterprise Architect, Papyrus etc.
- Course Project
  - Vital and major component of the course
  - Group of (3) students
- Weekly Task
  - Coding Tasks

Course Contents

Outline

1. Introduction to course and pedagogy
2. Logical architecture – 3 tier and N-tier architectures
3. Overview of OOP, OOAD and UML
4. Capturing User requirements via Use-Cases
5. Understanding problem domain-Domain Model
6. Analyzing actor actions – System sequence diagrams
7. Modeling dynamic software behavior using sequence diagrams
8. Design class diagrams
9. Designing for visibility (Mapping design to code)
10. Assigning roles and responsibilities using GRASP
11. Learning from industry experience – Gang of Four design patterns
12. Packaging and deploying software

Course Content

• OOP Basics • Three tier architecture • Requirements elicitation • Use cases	FIRST SESSION
• Domain modeling • Interaction modeling • GRASP • Behavior modeling • Gang of four patterns	SECOND SESSION
• Gang of four patterns • Logical Architecture • Design to code • Software Development tools and configuration management	THIRD SESSION

Textbooks and Supplementary Reading List

- Text Book(s)
  - Craig Larman, Applying UML and Patterns, 3rd edition, 2004.
  - Bernd Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering, using UML, Patterns, and Java.
  - UML 2 Toolkit Published by Wiley Publishing, Inc.
  - UML 2.0, Documentation:www.rational.com

- Additional Reading List
  - Unified Modeling Language: Superstructure, version 2.1.2, formal/07-11-02, OMG, www.omg.org

✓ Available in the Library/E-Book

Some Rules

- Raise your hand before asking any question
- Don’t miss a class
  - No retakes (except for Mid or Final Exam)
- Don’t “sleep” in the class
  - You might miss a quiz ☹
- Avoid using mobile phone in the class
- **Above all, whatever you do, please do not disturb others**

Attendance Policy

- Attendance will be taken at the beginning of the lecture
- **80% attendance is mandatory**
- Students coming late will be marked as **late**
- 2 late arrivals are equal to an absent

Dishonesty, Plagiarism

- All parties involved in any kind of cheating in (Quizzes, Assignments & Projects) will get ZERO in that exam
- Plagiarism in midterm/ final exam may result in F grade in the course
- Habitual cases will be awarded F

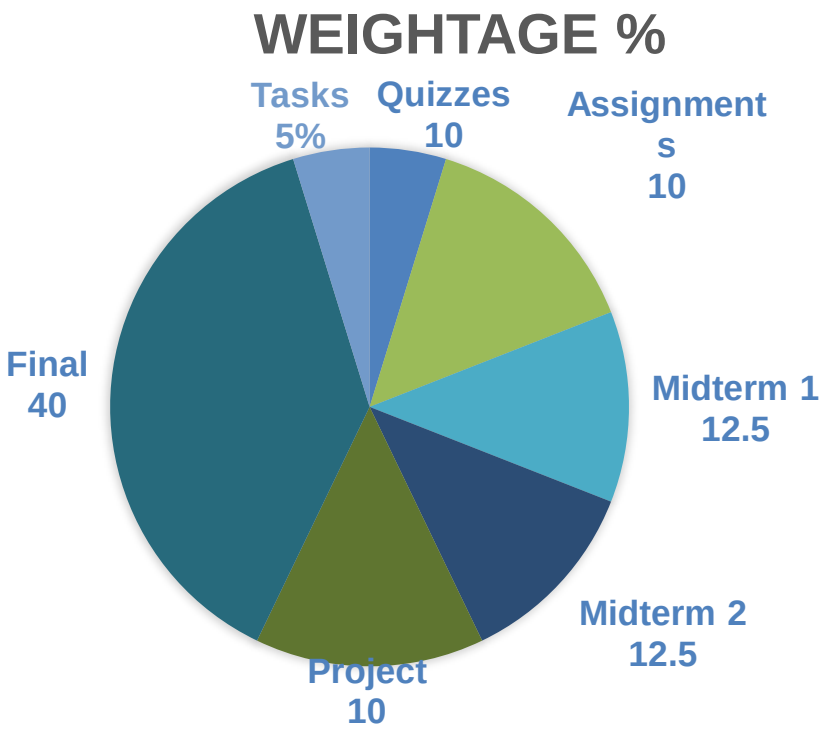
Google Classroom Code

y6dpncj

Evaluation Breakdown

- Grading Policy
  - Absolute Grading

- Theory breakdown



Revision

- Functional vs non-functional Requirements
- Difference between Agile, DevOps, Scrum
- What is software development life cycle?
- What are the phases of SDLC?
- What is SRS?
- Difference between Analysis and Design?
- What is reliability?

Software use during this course

- UML (Designing/Modeling Standard)
- Rational Software Architect (Modeling tool)
- Eclipse IDE (Development IDE)

Unified Modeling Language

- A de facto industry standard by the Object Management Group (OMG)
- Standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing object oriented software and the software development process.
- The UML uses mostly graphical notations to express the design of software projects.
- Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Overview of UML Diagrams

emphasizes the static structure of the system using objects, attributes, operations and relationships.

Structural

: element of spec. irrespective of time

- Class
- Component
- Deployment
- Object
- *Composite structure*
- *Package*

Behavioral

: behavioral features of a system / business process

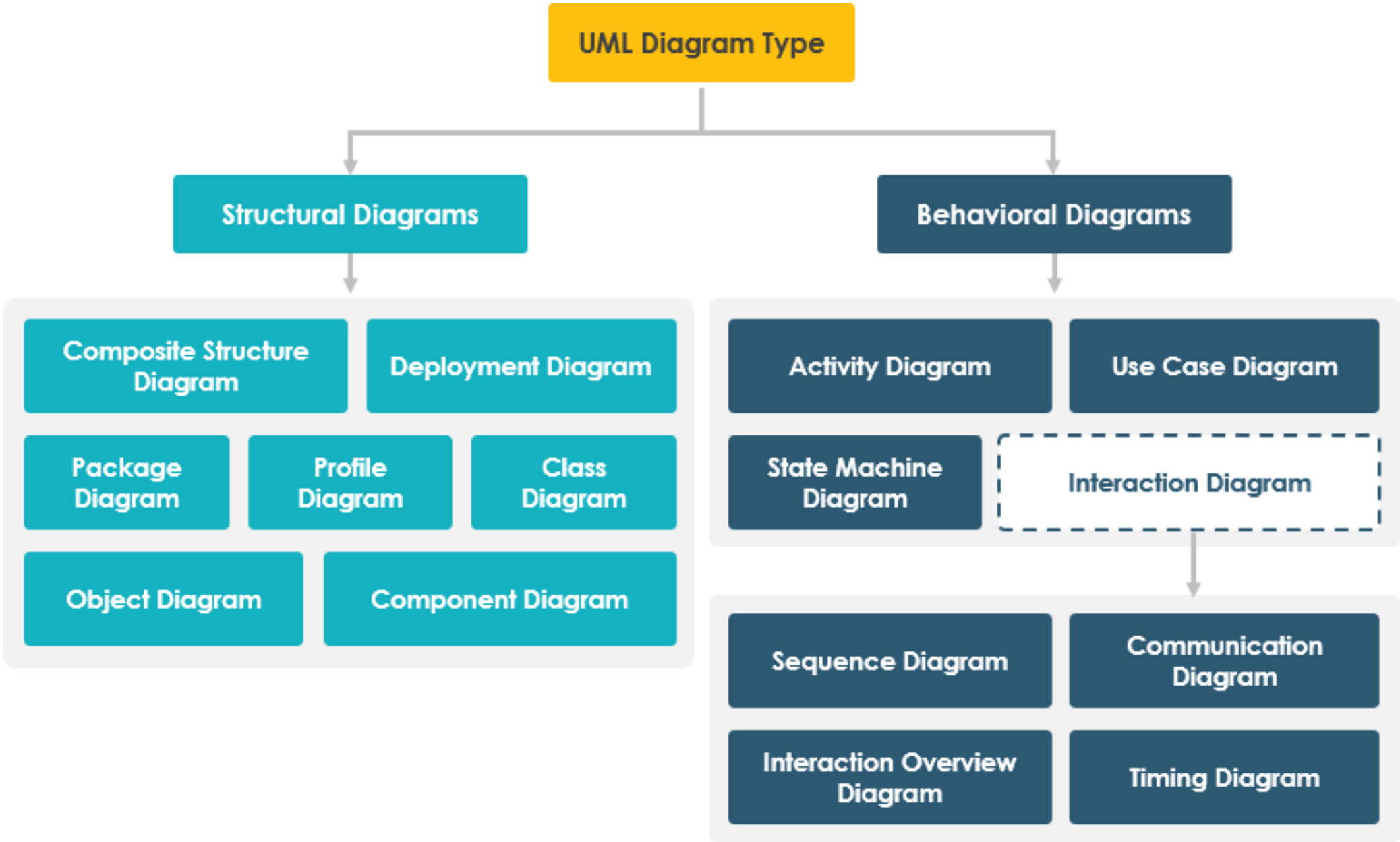
- Activity
- State machine
- Use case
- *Interaction*

Interaction

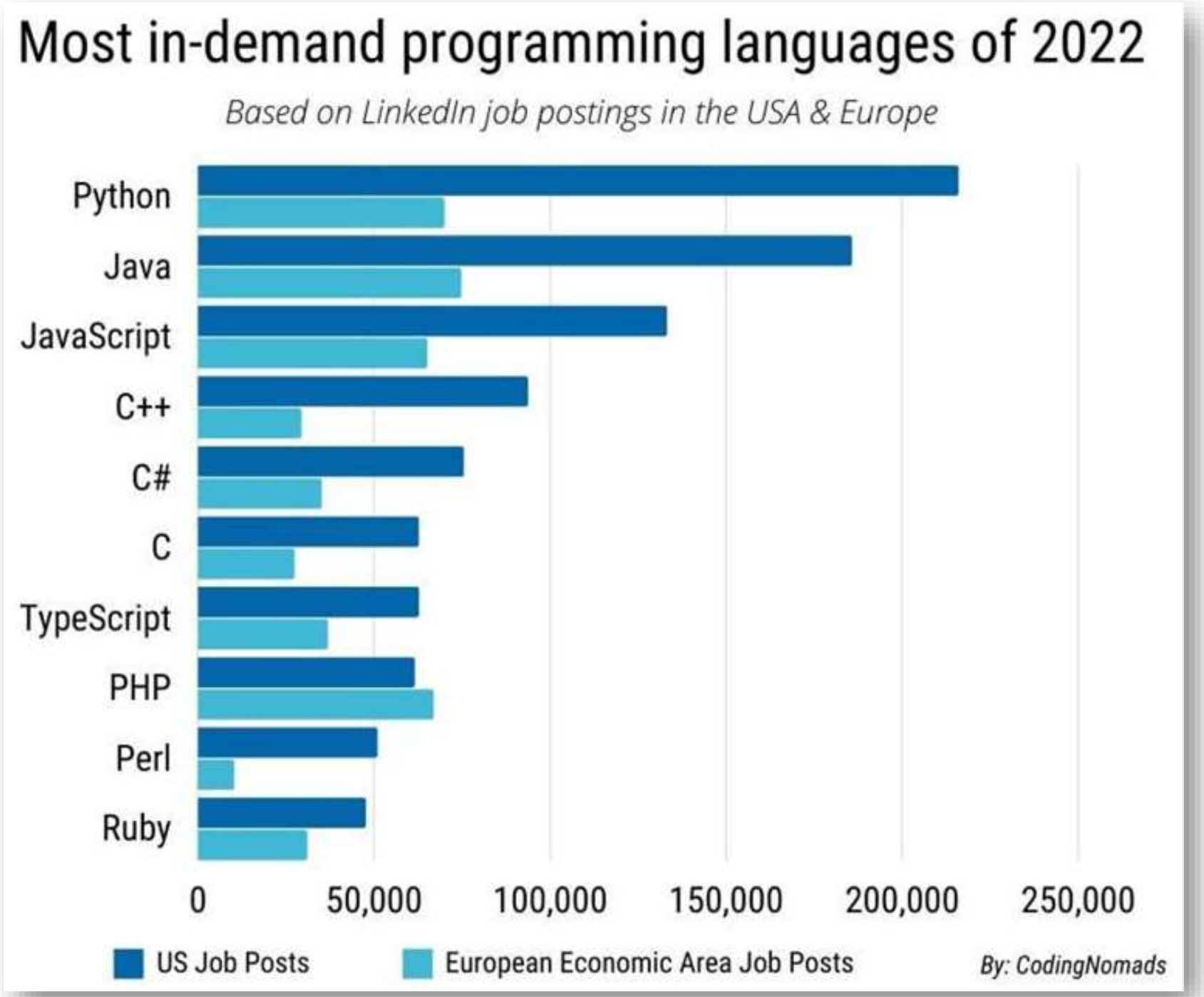
: emphasize object interaction

- Communication(collaberation)
- Sequence
- *Interaction overview*
- *Timing*

emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects.


UML Diagram Type
Structural Diagrams
Behavioral Diagrams
Composite Structure Diagram
Deployment Diagram
Package Diagram
Profile Diagram
Class Diagram
Object Diagram
Component Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Interaction Overview Diagram
Timing Diagram

Why Java?


Most in-demand programming languages of 2022
Based on LinkedIn job postings in the USA & Europe
Python
Java
JavaScript
C++
C#
C
TypeScript
PHP
Perl
Ruby
0
50,000
100,000
150,000
200,000
250,000
US Job Posts
European Economic Area Job Posts
By: CodingNomads

Source: spectrum.ieee.org/static/interactive-the-top-programming-languages-2020

"Java is the main language behind Android, which owns an 85% share of the mobile market. It's also the most popular language for Internet of Things (IoT) devices."

Java Spread

A lot of companies are using Java now-a-days, rather than other languages due to a simple fact: Java is awesomely **platform independent** and **reliable** which makes in famous in almost all fields including android, ios, smart TV, web development.

Motivation behind developing Java

"Write Once,
Run Anywhere."

What is Java?

- It is an object-oriented language developed by Sun in the mid 1990s.
  - Original language called Oak
  - Intended for embedded systems
  - Based on C and C++
  - Now, popularly used in console, GUI, web and mobile applications

- Sun describes it as
  - "A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multi-threaded and dynamic language."

What was the reason of creation?

- Need a cross-platform language for multiple-device development.
- C++ though object oriented and widely used was highly resource intensive.
- Developed with the aim of “**Write Once, Run Anywhere**”.

What is Java? (cont)

- Object-Oriented Programming Language
  - Designed to support Object-Oriented concepts
- Platform
  - To execute java on your machine, you need java runtime environment (JVM)
  - JVM accept the byte code and executes the machine.
- Distributed
  - Applications are constructed using objects. Objects can be distributed in multiple locations within a network environment.
  - Extensive integration with TCP/IP
- Interpreted
  - Java compiles to byte-code (not machine code). Byte code is interpreted.
  - Most Java versions after 1.2 include a JIT (Just-In-Time) compiler which compiles byte code to machine code.

What is Java? (cont)

- Robust
  - Memory management is done automatically
  - Exception handling support
- Secure
  - All Java code subject to security model.
  - JVM verify that byte code and ensure that it is not performing any unsafe action.
  - No pointer so access to arbitrary memory location is restricted.
- Architecture-Neutral/Portable
  - Compiled Java (byte code) will run on any platform which has a Java Virtual Machine
  - The Java Virtual Machine is available for almost all platforms...
    - Even mainframes.

What is Java? (cont)

- Multi-Threaded
  - Processes contain multiple threads of execution.
  - Similar to multi-tasking but all threads share the same memory space.

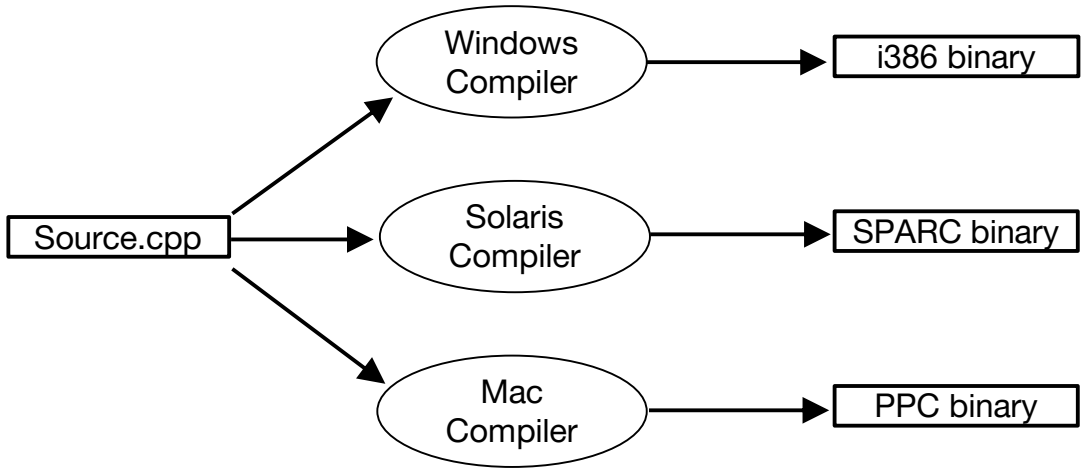
- Rich set of Libraries
  - The Java API is extensive.
  - The standard JDK comes with over 200 built-in packages containing Java APIs

Platform Independence. How does Java do it?

- Java has been described as WORA (Write once, Run Anywhere)
- Because Java source code is compiled to byte code and the byte code is interpreted, Java code can be executed anywhere an interpreter is available.
- The "Interpreter" is call the Java Virtual Machine

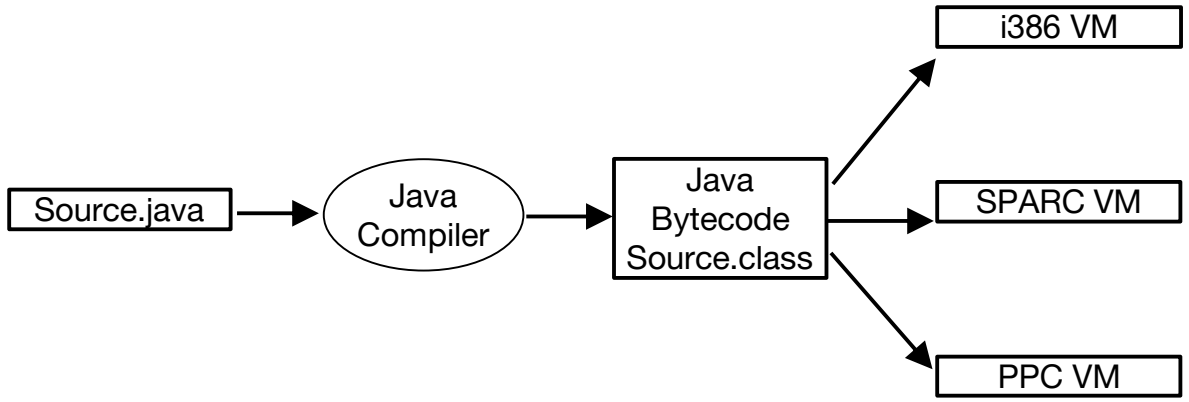
The Java Virtual Machine.

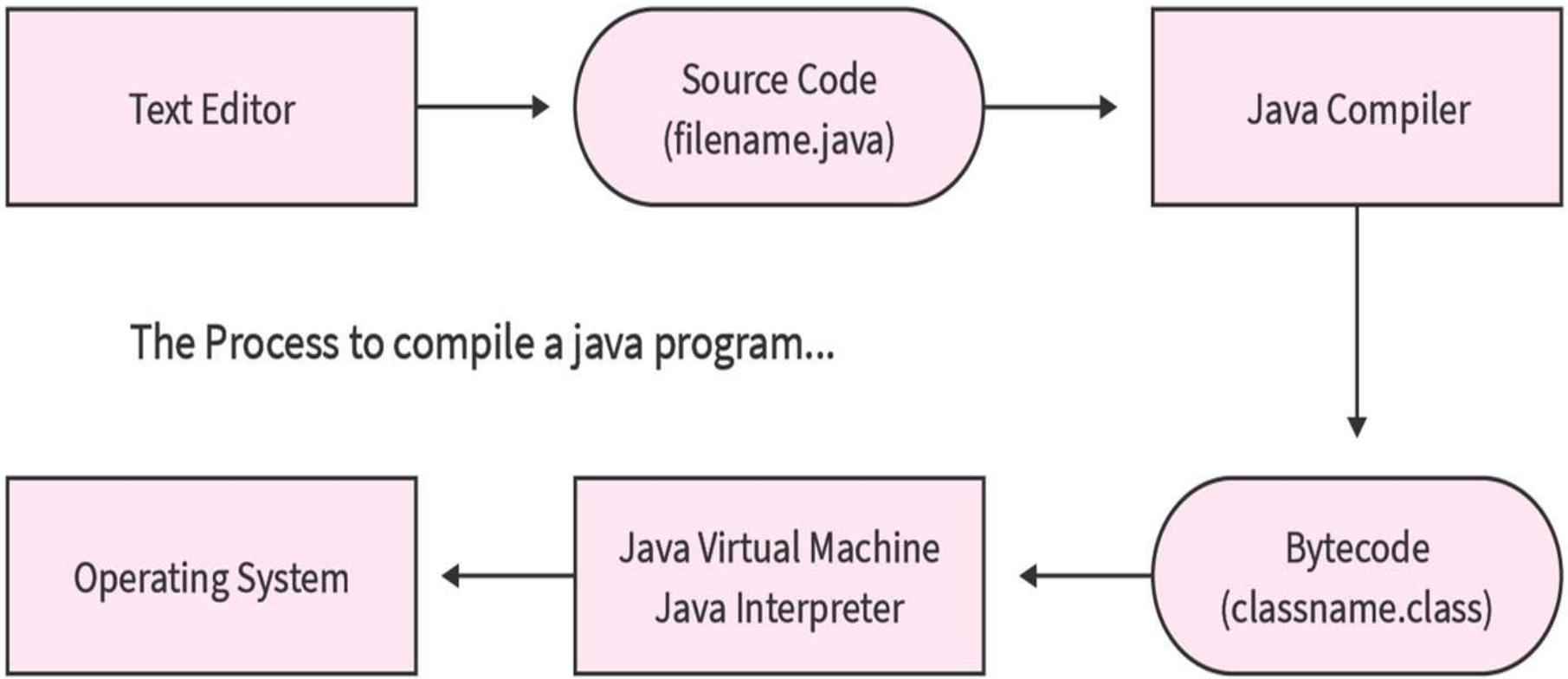
- Traditionally, source code had to be compiled for the target hardware and OS platform:

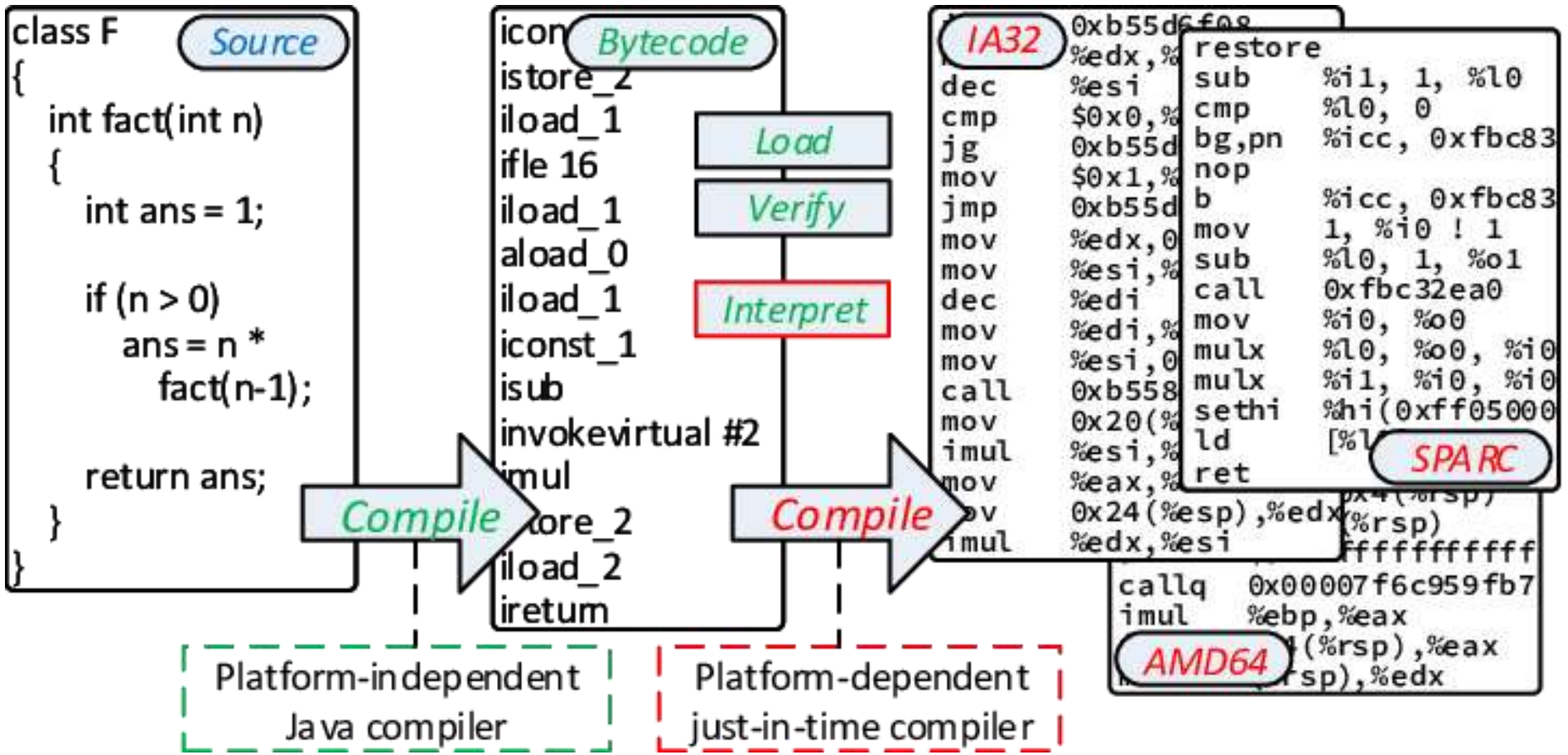


The Java Virtual Machine.

- Java source files (.java) are compiled to Java bytecode (.class)
- Bytecode is interpreted on the target platform within a Java Virtual Machine




Text Editor
Source Code
(filename.java)
Java Compiler
The Process to compile a java program...
Operating System
Java Virtual Machine
Java Interpreter
Bytecode
(classname.class)


class F
{
int fact(int n)
{
int ans = 1;
if (n > 0)
ans = n *
fact(n-1);
return ans;
}
}
Source
Compile
Platform-independent
Java compiler
Bytecode
istore_2
iload_1
ifle 16
iload_1
aload_0
iload_1
iconst_1
isub
invokevirtual #2
imul
iload_2
ireturn
Load
Verify
Interpret
Compile
Platform-dependent
just-in-time compiler
IA32
dec %esi
cmp $0x0,%
jg 0xb55d
mov $0x1,%
jmp 0xb55d
mov %edx,0
mov %esi,%
dec %edi
mov %edi,%
mov %esi,0
call 0xb558
mov 0x20(%
imul %esi,%
mov %eax,%
mov 0x24(%esp),%edx
imul %edx,%esi
callq 0x00007f6c959fb7
imul %ebp,%eax
AMD64
restore
sub %i1, 1, %l0
cmp %l0, 0
bg,pn %icc, 0xfbc83
nop
b %icc, 0xfbc83
mov 1, %i0 ! 1
sub %l0, 1, %o1
call 0xfbc32ea0
mov %i0, %o0
mulx %l0, %o0, %i0
mulx %i1, %i0, %i0
sethi %hi(0xff05000
ret
SPARC

The Java Software Development Kit (SDK)

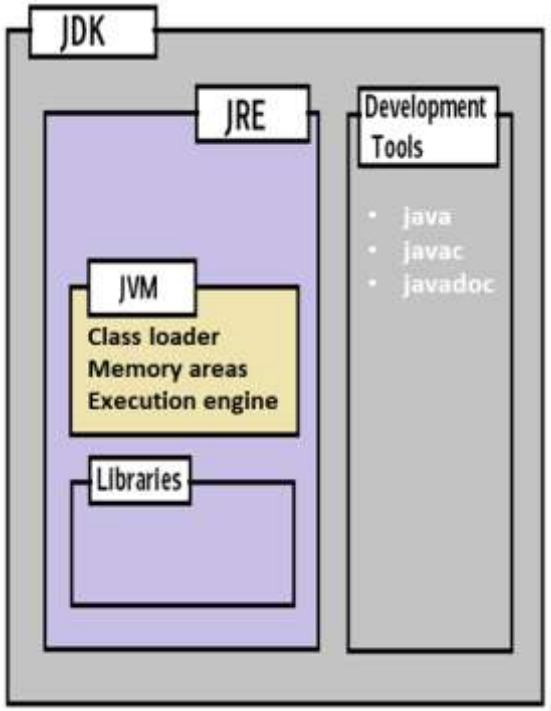
- The Java SDK comes in three versions:
  - J2ME - Micro Edition (for handheld and portable devices)
  - J2SE - Standard Edition (PC development)
  - J2EE - Enterprise Edition (Distributed and Enterprise Computing)

Common Terms

1. **JDK** – **Java Development Kit** (in short JDK) is Kit which provides the environment to **develop and execute(run)** the Java program. JDK is a kit(or package) which includes two things
   1. Development Tools (to provide an environment to develop your java programs)
   2. JRE (to execute your java program).

2. **JRE** – **Java Runtime Environment** (to say JRE) is an installation package which provides environment to **only run(not develop)** the java program(or application)onto your machine. JRE is only used by them who only wants to run the Java Programs i.e. end users of your system.

1. **JVM** – **Java Virtual machine**(JVM) is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for **executing the java program line by line** hence it is also known as interpreter.



HelloWorld.java

- Here is Java's "HelloWorld" implementation:

In the file, HelloWorld.java:

```
public class HelloWorld
{
  public static void main(String[] args)
  {
        System.out.println("Hello World");
  }
}
```

COURSE PROJECT

Should be MIS

SW Development Project: Deliverable 1

- Deliverable #1:
  - **Deadline: (29th August)**
- COMPANY
  - Company Name (of your newly established company)
  - Company Logo
- TEAM (3) Members
  - (Team) Manager
  - (Requirement) Analyst
  - Manager QA
  - Designer
  - Developers/SW Engineers

1	APEX
2	SYNOPSIS
3	Chimera Soft
4	LOGIX
5	Envision Tech
6	HEDZ
7	Consumate Logiciels
8	Logistics
9	ATEK Inc
10	Pi Sigma
11	Erap Solutions
12	Mantaq Solutions
13	Design Minds
14	Dream Technologies
15	Logic All
16	Synergy
17	Solution Providers
18	Ferocity Inc
19	Cryph Tech
20	COGILENT

SW Development Project: Deliverable 2

- **Deliverable #2**
  - Project Selection
    - Meeting with the client
    - Project Proposal
    - Present the idea of the project and what is aimed to achieve
  - Sections
    - Project Title
    - Scope
    - Objectives
    - Problem Statement and Description

Project Proposal Components

- Title
  - Should be well defined Indicate the project aim and immediate solution its going to provide.
- Scope
  - Defines the domain of the project. List any previous tools or work done in the same area. Also identify the level to which the project is going to be probed into in future.
- Objective
  - Outline the major aims/goals that are aimed to be achieved through the project
- Problem Statement and Description
  - Explain the major reason for choosing the project and the problem addressed by the project. Describe the problem in detail (e.g. office automation mitigates manual work etc)
  - Should not be more than 2-3 paragraphs
  - Can also explain Feasibility.

SW Development Project: Deliverable 2

- **Deadline: 5th Sep, 2022**
  - Project Title
  - Scope
  - Objectives
  - Problem Statement and Description
- Project Proposal and other deliverables (to be explained later on by Course Instructors and LI's)
- Deliverable format : Found in course book, explained by course instructor OR explained in the assignment description.
- If in doubt, YOUR responsibility to ask AHEAD of deadline, not ON or AFTER IT!

Home Work

- Please revise OOP principles

Thanks

National University
Of Computer and Emerging Sciences

quest lab