# Object Oriented Systems Development Life Cycle

# Object Oriented Systems Development Life Cycle

- **System development** can be viewed as a process
- The **developmen**t itself, ***is a process of change***, refinement, transformation, or addition to existing product.
- The **process** can be divided into small, interacting phases – sub processes. Each sub process must have the following:
  - A **description** in terms of how it works
  - **Specification** of the **input** required for the process
  - **Specification** of the **output t**o be produced

# Contd…

- The **software development process** can be viewed as a **series of transformations**, where the output of one transformation becomes the input of the subsequent transformation.
- **Transformation 1 (analysis)** – translates the users' needs into system requirements and responsibilities.
- **Transformation 2 (design)** – begins with a problem statement and ends with a detailed design that can be transformed into an operational system.

# Contd…

- **Transformation 3 (implementation)** – refines the detailed design into the system **deployment** that will satisfy users’ needs.
- An **example** of the **software development process** is the **waterfall approach**, which starts with deciding what is to be done. Once the requirements have been determined, we next must decide how to accomplish them. This is followed by a step in which we do it, whatever “it” has required us to do. We then must test the result to see if we have satisfied the users’ requirements. Finally, we use what we have done.

# Contd…

- **To achieve high quality in software we should be able to answer the following questions:**
- How do we determine that the system is ready for delivery?
- Is it now an operational system that satisfies users’ needs?
- Does it pass an evaluation process?
- **describes** a means of system evaluation in terms of **four quality measures**:

# Contd…

- **Correspondence –** measures how well the delivered system matches the needs of the operational environment, as described in the original requirements statement.
- **Validation –** task of predicting correspondence.
- **Correctness –** measures the consistency of the product requirements with respect to the design specification.
- **Verification –** exercise of determining correctness.

# Contd…

- **Validation begins** as soon as the **project starts**, but **verification** can begin only after a **specification has been accepted.**
- The object-oriented software development life cycle (SDLC) consists of three macro processes:
  - object-oriented analysis
  - object-oriented design
  - object-oriented implementation.