Benazir Bhutto Shaheed University Lyari, Karachi

Department of Computing Science & Information Technology

Object Oriented Programming (CS and IT-321)
2nd Semester

Object Oriented Design Concepts

Program Development

- The creation of software involves four basic activities:

  - establishing the requirements

  - creating a design

  - implementing the code

  - testing the implementation

- These activities are not strictly linear – they overlap and interact

Requirements

- *Software requirements* specify the tasks that a program must accomplish

  - <u>what</u> to do, not how to do it

- Often an initial set of requirements is provided, but they should be critiqued and expanded

- It is difficult to establish detailed, unambiguous, and complete requirements

- Careful attention to the requirements can save significant time and expense in the overall project

Design

- A *software design* specifies <u>how</u> a program will accomplish its requirements

- That is, a software design determines:

  - how the solution can be broken down into manageable pieces

  - what each piece will do

- An object-oriented design determines which classes and objects are needed, and specifies how they will interact

- Low level design details include how individual methods will accomplish their tasks

Implementation

- *Implementation* is the process of translating a design into source code

- Novice programmers often think that writing code is the heart of software development, but actually it should be the least creative step

- Almost all important decisions are made during requirements and design stages

- Implementation should focus on coding details, including style guidelines and documentation

Testing

- *Testing* attempts to ensure that the program will solve the intended problem under all the constraints specified in the requirements

- A program should be thoroughly tested with the goal of finding errors

- *Debugging* is the process of determining the cause of a problem and fixing it

- We revisit the details of the testing process later in this chapter

Object Oriented Programming

Problem Solving

Definition

- It is a systematic approach to find and implement the solution of a problem.

Steps in Problem Solving

- Problem Definition
- Problem Analysis
- Design
- Coding
- Testing
- Maintenance

Problem Definition

- To solve a problem, the first step is to identify and define the problem.

- The problem must be stated clearly, accurately and precisely.

For example: Find largest of three numbers

Problem Analysis

- The problem analysis helps in designing and coding for that particular problem.

1. Input specifications

The number of inputs and what forms the input are available

2. Output specifications

The number of outputs and what forms the output should be displayed.

E-x input – a,b,c

output - c

Designing a program

- 1. Algorithms
- 2. Flowcharts
- Algorithm - step by step procedure of solving a problem
- Flowcharts – It is the graphical representation of the algorithm.

Coding

- Writing instructions in a particular language to solve a problem.

Testing a Program

- After writing a program, programmer needs to test the program for completeness, correctness, reliability and maintainability.

- Unit testing

- Program Testing

- Verification Testing

- Validation Testing

Maintaining the program

- It means periodic review of the programs and modifications based on user requirements.

Algorithm

Algorithms

- The word *algorithm* comes from the name of a Persian mathematician Abu Ja'far Mohammed ibn-i Musa al Khowarizmi.

- In computer science, this word refers to a special method useable by a computer for solution of a problem. The statement of the problem specifies in general terms the desired input/output relationship.

- For example, sorting a given sequence of numbers into nondecreasing order provides fertile ground for introducing many standard design techniques and analysis tools.

Cont.

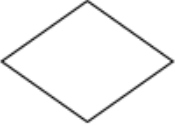A sequential solution of any program/project that written in human language, called algorithm.

Algorithm is first step of the solution process, after the analysis of problem, programmer writes the algorithm of that problem.

An algorithm should always have a clear stopping point.

Flowcharts

- Graphically depict the logical steps to carry out a task and show how the steps relate to each other.

Flowchart symbols

Symbol	Name	Meaning
→	*Flowline*	Used to connect symbols and indicate the flow of logic.
(rounded rectangle)	*Terminal*	Used to represent the beginning (Start) or the end (End) of a task.
(parallelogram)	*Input/Output*	Used for input and output operations, such as reading and displaying. The data to be read or displayed are described inside.
(rectangle)	*Processing*	Used for arithmetic and data-manipulation operations. The instructions are listed inside the symbol.
(diamond)	*Decision*	Used for any logic or comparison operations. Unlike the input/ouput and processing symbols, which have one entry and one exit flowline, the decision symbol has one entry and two exit paths. The path chosen depends on whether the answer to a question is "yes" or "no."

Pseudocode

- Uses English-like phrases to outline the task.
- pseudocode is a method of developing an algorithm. Programmers can use informal simple language to write a pseudocode and there is no strict syntax to follow

Pseudocode example

Determine the proper number of stamps for a

letter

Read Sheets *(input)*

Set the number of stamps to Sheets / 5 *(processing)*

Round the number of stamps up to the next

whole number *(processing)*

Display the number of stamps *(output)*

Coding

- Coding is the translation of an algorithm or flowchart into a suitable computer language c,c++,java

Testing and Debugging

- To achieve the required output, the program that is written in coding must be tested ,compiled and executed.
- Types of errors

  syntax error

  semantic error

  Run-time error

Debugging

- It is the process of identifying and correcting the bugs.

Maintenance

- Periodic review of the program and modifications based on their user requirements.