Chapter 7:

Programming Guidelines

Informatics Practices

Class XI (CBSE Board)

Revised as per CBSE Curriculum 2015

"Open Teaching-Learning Material"

Visit www.ip4you.blogspot.com for more….

Authored By:- Rajesh Kumar Mishra, PGT (Comp.Sc.)
Kendriya Vidyalaya Upper Camp, Dehradun (Uttarakhand)
e-mail : rkmalld@gmail.com

Objective

In this presentation you will learn about the various guidelines regarding development of a good program.

- Understanding the goodness of a program
- Stylistics Guidelines of a Program.
- Guidelines for using GUI controls.
- Program Development phases.
- Programming Errors and their types.

Characteristics of a good Program

A good Program not only efficiently solves the given problem but also it should be presentable. The following features are expected for its goodness-

- □ **Effective & Efficient**

  It must produce correct result for what it is written. The program should be time and memory efficient.

- □ **User Friendly**

  The Interface (form Design) should be pleasant, attractive and well labeled, so that user can handle it effectively. The output produced by the program should also understandable and presentable.

- □ **Self- Documenting Code**

  The source code of the program should have meaningful names for Identifiers and Variables, and proper comment should be used to make it self-documenting.

- □ **Reliable**

  The Program should be reliable i.e. able to handle errors and give accurate result.

- □ **Portable**

  A Program should be portable i.e. It can run on different Hardware and software platform.

Stylistic Guidelines- How to write a good program

Writing a good Program is an art. You should follow the following points while writing a program-

- □ **Meaningful Names for Identifiers**

  Variable, Class and Method names should be meaningful and unambiguous. It makes the clarity and readability of the program.

- □ **Ensure clarity of expression**

  Expression in the program should be written clearly. Proper parenthesis may be used to enhance the clarity of the expression.

- □ **Use proper Comment & Indentation**

  A Program should have proper comment to explain the various parts of the program. Proper Indentation makes the program more readable.

- □ **Insert Blank Lines and spaces**

  Blank lines can inserted before the declaration of blocks, comment lines, method's prototype etc. It increases readability. Space must be given before and after an operator to make it more readability.

- □ **Statement Formatting**

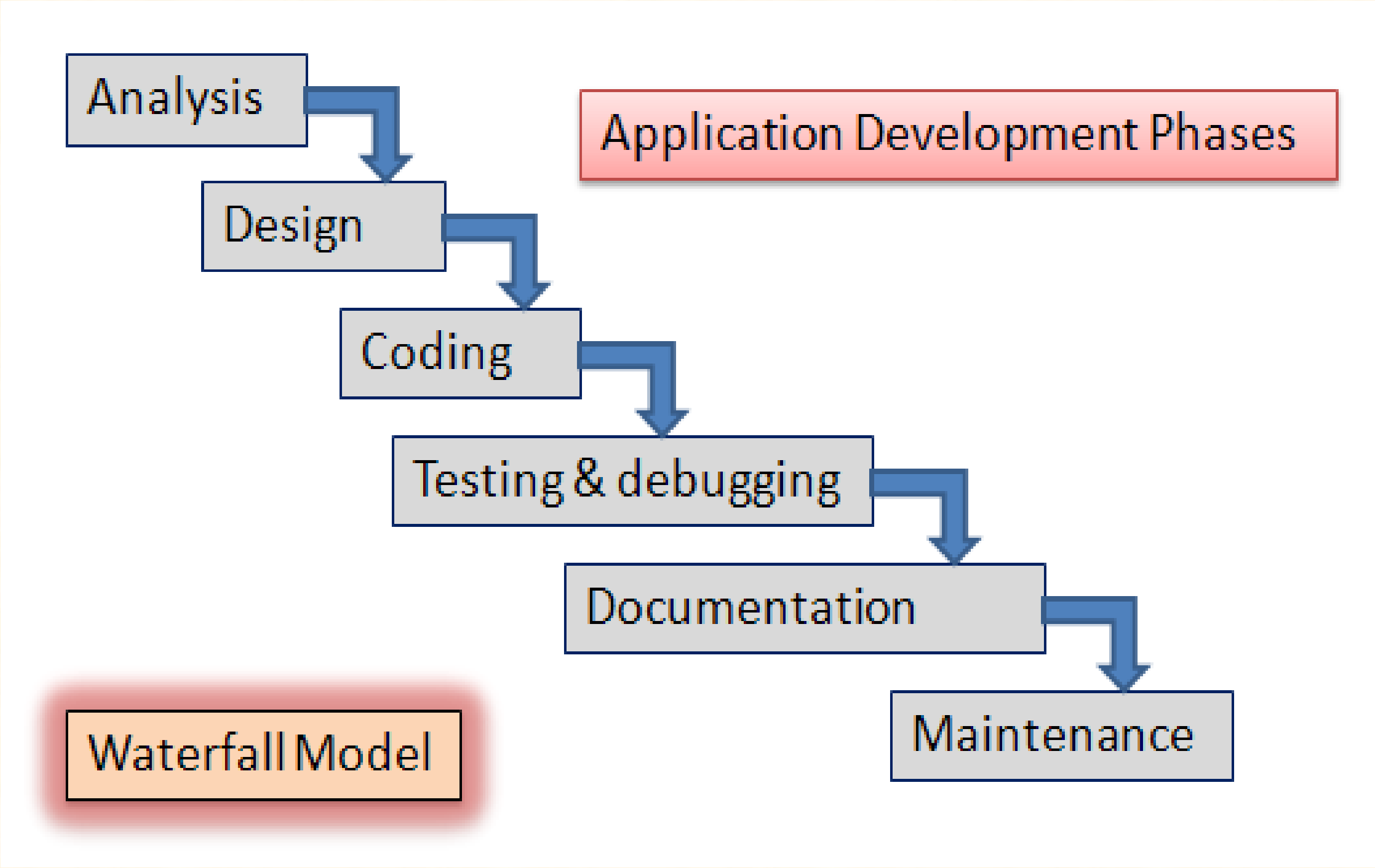
  Program should be pretty typed i.e. All the statement must be written in separate lines. So that program may be readable and understandable.

Guidelines for using GUI Controls

- Make sure that the user provides appropriate information with minimum efforts. So radio buttons, checkboxes, combo boxes, and lists should be used at maximum level instead of textboxes.
- Radio Button should be used wherever one of the option out of limited number of known set of options are required to be selected. For example, for accepting gender (Male or Female), marital status (Single or Married), for accepting membership type etc.
- Checkbox should be used wherever multiple options are required to be selected from a limited number of known set of options. For example, for accepting food order in a restaurant (Pizza, Burger, Pao Bhaji, Chowmein) etc.
- Combo box should be used wherever only one of the options from a large number of known set of options is required to be taken from the user. For example, selecting state and selecting city etc.
- List should be used wherever multiple options are required to be selected from a large number of known set of options. For example, selecting multiple food items from a menu containing multiple items.
- It is advisable to use appropriate labels for each input and output options to help the user for correct interpretation.

Program Development Phases(steps)

A program development process is a step by step process where each step (stage) contributes to building of an effective and efficient program. In general, the following steps are taken to develop an application.



How to develop a Program (Steps)

□ Analysis of the Problem

This phase involves the following steps:

- ❖ You must recognize the need of program. Try to explore ‘**WHAT**’ component in the problem
- ❖ After determining the need, carefully analyze the problem to get an effective solution. Try to explore the inputs to get required output.

□ Design Program

In this step ‘**HOW**’ component is explored. This phase involves planning of step-by-step procedures required to solve a given problem. At this stage a detailed design of the following components is to be completed:

Input Design: It involves defining the data items to be entered by the user into the application with minimum ambiguity and inconsistency.

Output design: Decide how, where and when result or information it is to be displayed.

User Interface (Forms) Design: It involves designing of the screen (interface forms) to accept input given by user and display data/information to user.

Process design: The big program is divided into smaller components called module or Process. Each programming module handles a specific task.

Database Design: involves creating database and its table which to be used for storing user input in the form of records. Database plays very important role and it should be designed carefully to avoid ambiguity and redundancy.

How to develop a Program …..

- □ **Coding of the Program**

  This phase involves actual writing of programs using appropriate programming languages. In this phase, Solution-steps of the problem is translated in to actual Program using an IDE. A good programmer will make an optimum code, which is readable, easy to understand and maintain with appropriate error handling, comments and indentation. For example, after designing the form, we wrote the actual code using Java's IDE NetBeans.

- □ **Testing & Debugging of the program**

  Once a Program has been coded, it can be executed to find out the errors (Syntactical & Logical), if any. The errors also termed as 'bugs'. The rectifying errors is called 'debugging'. Testing is done to find out the errors (bugs) but debugging is used to remove the errors. In the testing and debugging stage, we should try out all possible inputs in order to make our application error free. There are various approaches for testing and application like-

  - ❖ Functional (black box) & Structural (White box) Testing
  - ❖ Module (unit) & Integration testing.

How to develop a Program …..

- □ **Documentation**

Documentation refers to written description of I/O design, source code and, internal and external requirement to the program. Documentation makes a program more understandable, readable and more easily modifiable.

A Documentation describe the **who**, **what**, **when**, **how** and **where** of the system or application. The documentation module (also called Manual) may be following types as per their contents.

User Manual:

Describes how to use the program and contains all the I/O screens and operating instruction for its user.

Program Manual:

It contains the Source code of the program for reference purpose. It is required during program Maintenance.

System Manual:

It contains overall specification, results of all the program development stages etc.

How to develop a Program …..

- □ **Maintenance of the Program**

  Program maintenance refers to the modification of a program, in order to meet the following-

  - ✓ to remove the undetected errors or
  - ✓ to enhance the program functionality
  - ✓ to keep the program up-to-date with its environment.

The following types of Maintenance may occur in the application-

Corrective Maintenance: It is required to correct the errors which were undetected during the design of the program.

Adaptive Maintenance: Change in user's need or program environment may result modification in the program to adapt such changes. This change in the program is called Adaptive Maintenance.

Preventive Maintenance: Sometimes a program is modified to avoid anticipated errors. Preventive Maintenance is carried out to avoid such errors.

Perfective Maintenance: To cope with changing world i.e. changes in the technology, you should tune your system to keep it up-to-date. Due to arrival of new technology and facilities, the program is updated to make it perfect in respect of new technological need.

Types of errors in the program

An error or bugs may occur while designing and testing of the application. These errors may prevent a program to run in correct manner, so error must be rectified before finalizing the application for the end-user. The following types are errors are found while testing a program.

□ Compile Time Error :

These types of errors occur during compile time. When a program is compiled, its sources code is checked for rules of programming language by the compiler. These errors are easy to locate and rectify. These are classified in two types-

- **Syntax Error:**

  Formal set of rules defined for writing any statement is known as syntax. Syntax errors occur when rules of the programming language are not followed. Some of the common examples of syntax errors are spelling errors in keywords, missing semicolon, missing parenthesis and using incompatible data types.

- **Semantic Error :**

  Semantic occurs when statement are not meaningful and ambiguous for compiler. For example if you are writing an expression x+y=z instead of z=x+y , then Semantic error occurs. This error is also located by the compiler.

Types of errors in the program

□ Run Time error:

Sometime an application looks syntactically correct and compiled well, but error occurred during execution time (Run time). These errors may cause abnormal termination of the application.

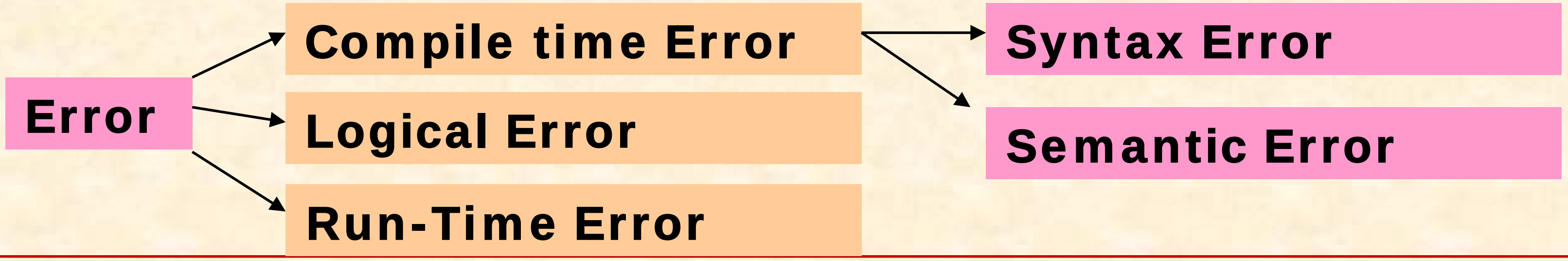
Run time errors are also called **exceptions**, and handling such errors is called exception handling. In Java exception-handling is done using **try{ }** and **catch{ }** blocks.

Example: Division by zero, trying to convert to number (int or double) from an empty jTextField etc. are example of run time error.

□ Logical Error :

Sometime a Program is free from compilation or runtime error but does not give correct result due to mistake in the calculation or use of operators etc. the it is called Logical Error. These errors are most difficult to trace in a big sized program.

Example: Suppose , if you using a wrong formula for calculating average of 3 subject marks, as Eng+Maths+GK/3 instead of (Eng+Maths+GK)/3 , you may get incorrect result.



Error

Compile time Error

Logical Error

Run-Time Error

Syntax Error

Semantic Error