CHAPTER 12: COMPUTER PROGRAMMING

12.1 Introduction

This unit specifies competencies required to develop computer program. It involves Identifying program and programming concepts, identifying phases of program development, perform program design and Analysis, develop a Computer program, Perform Program testing and debugging, Perform User training and Program Maintenance.

12.2 Performance Report

- Identified types of programming languages and concepts
- Identified Approaches of program development
- Identified Phases of program development
- Identified Program design and Analysis tools
- Identified Format of a computer program
- Adopted well written and readable programs using disciplined coding styles and standards
- Developed Maintenance schedule
- Determined Maintenance tools and techniques

12.3 Learning Outcome

12.3.1 List of the Learning Outcomes

These are the key learning outcomes, which make up workplace function:

- Identify program and programming concepts
- Identify Phases of Program development
- Perform program design and Analysis
- Develop a Computer program
- Perform Program testing and debugging
- Perform user training and Program Maintenance

12.3.2 Learning Outcome 1: Identify program and programming concepts

12.3.2.1 Learning Activities

The following are the performance criteria:

- Definition of program and programming is done
- Types of programming languages are identified
- Programming concepts are identified
- Approaches of program development are identified

Trainees to demonstrate knowledge in relation to:

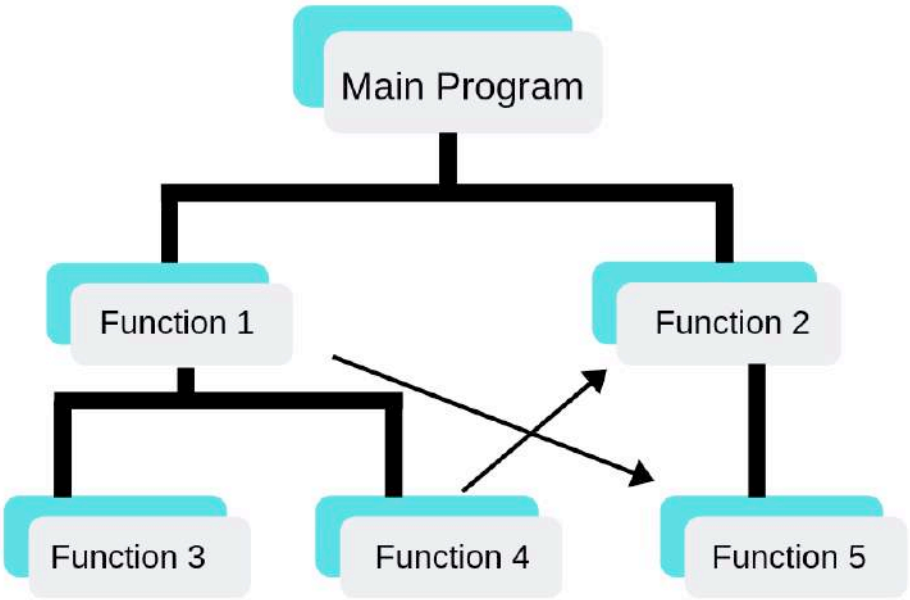
- Definition of program and programming
- Programming concepts
- Program structure: Variable declaration, Looping structures, Control structures, Syntax
- Programming languages: Object oriented, Functional, Imperative, Declarative
- Approaches of program development: Waterfall, Agile, Spiral etc

12.3.2.2 Information Sheet

A computer program is a collection of instructions that performs a specific task when executed by a computer. Most computer devices require programs to function properly.

A programming language is a set of commands, instructions, and other syntax use to create a software program.

Program structure The overall form of a program, with particular emphasis on the individual components of the program and the interrelationships between these components.



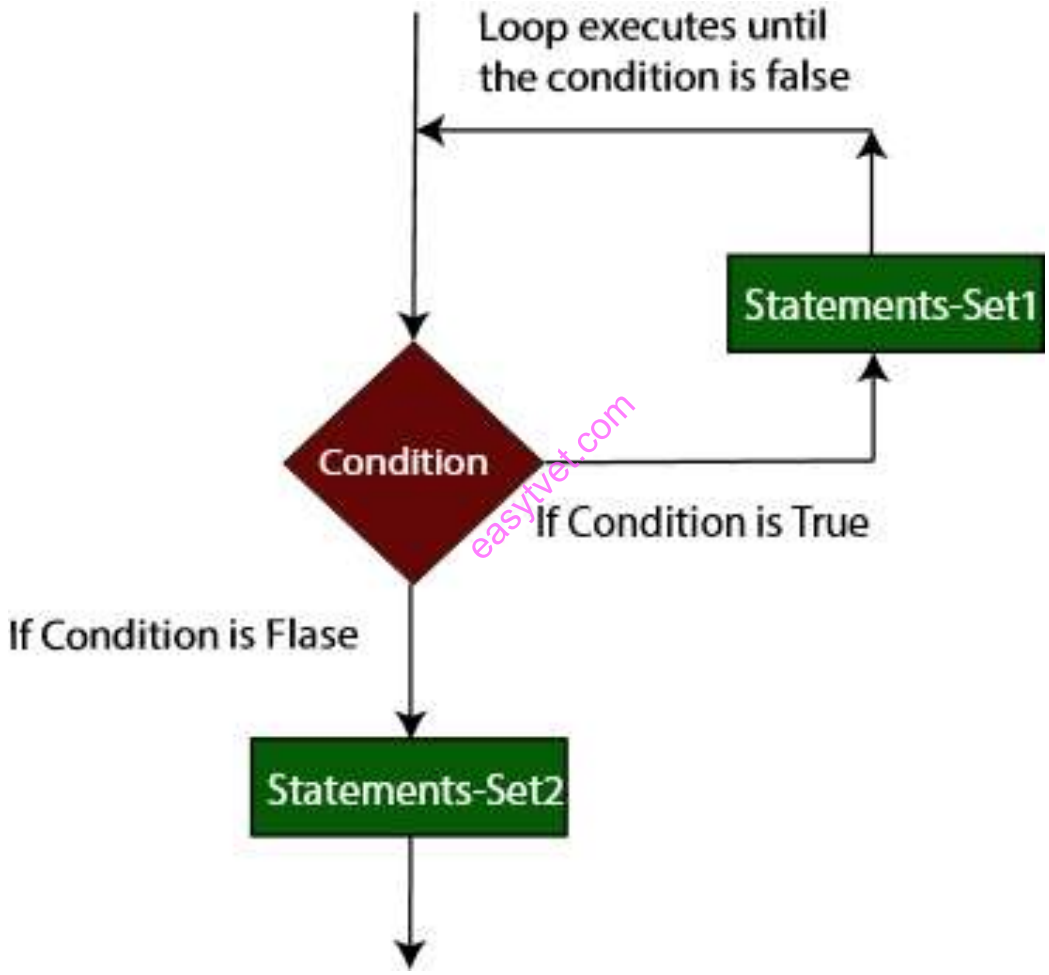
Source: www.codeavail.com

Figure 89: Program structure

A declaration of a variable is where a program says that it needs a variable. For small programs, place declaration statements between the two braces of the main method. The declaration gives a name and a data type for the variable. It may also ask that a particular value be placed in the variable.

Read: Declaration of variable: https://chortle.ccsu.edu/Java5/Notes/chap09A/ch09_3.html

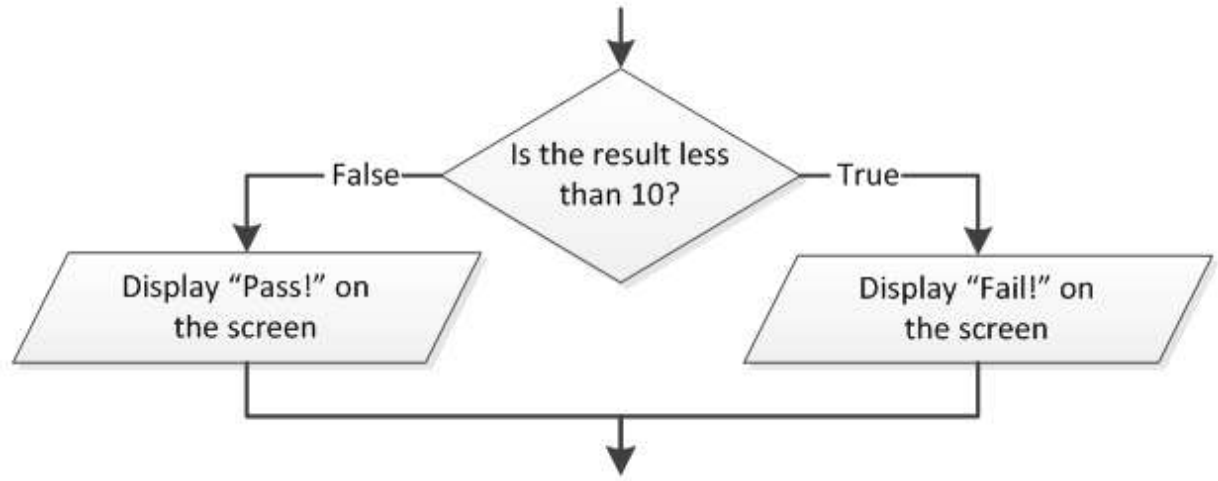
Loops are control structures used to repeat a given section of code a certain number of times or until a particular condition is met.



Source: www.tutorialscampus.com

Figure 90: Loops

A control structure is a block of programming that analyses variables and chooses a direction in which to go based on given parameters.



Source: https://www.bouraspage.com

Figure 91: Control structure

The syntax of a computer language is the set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language.

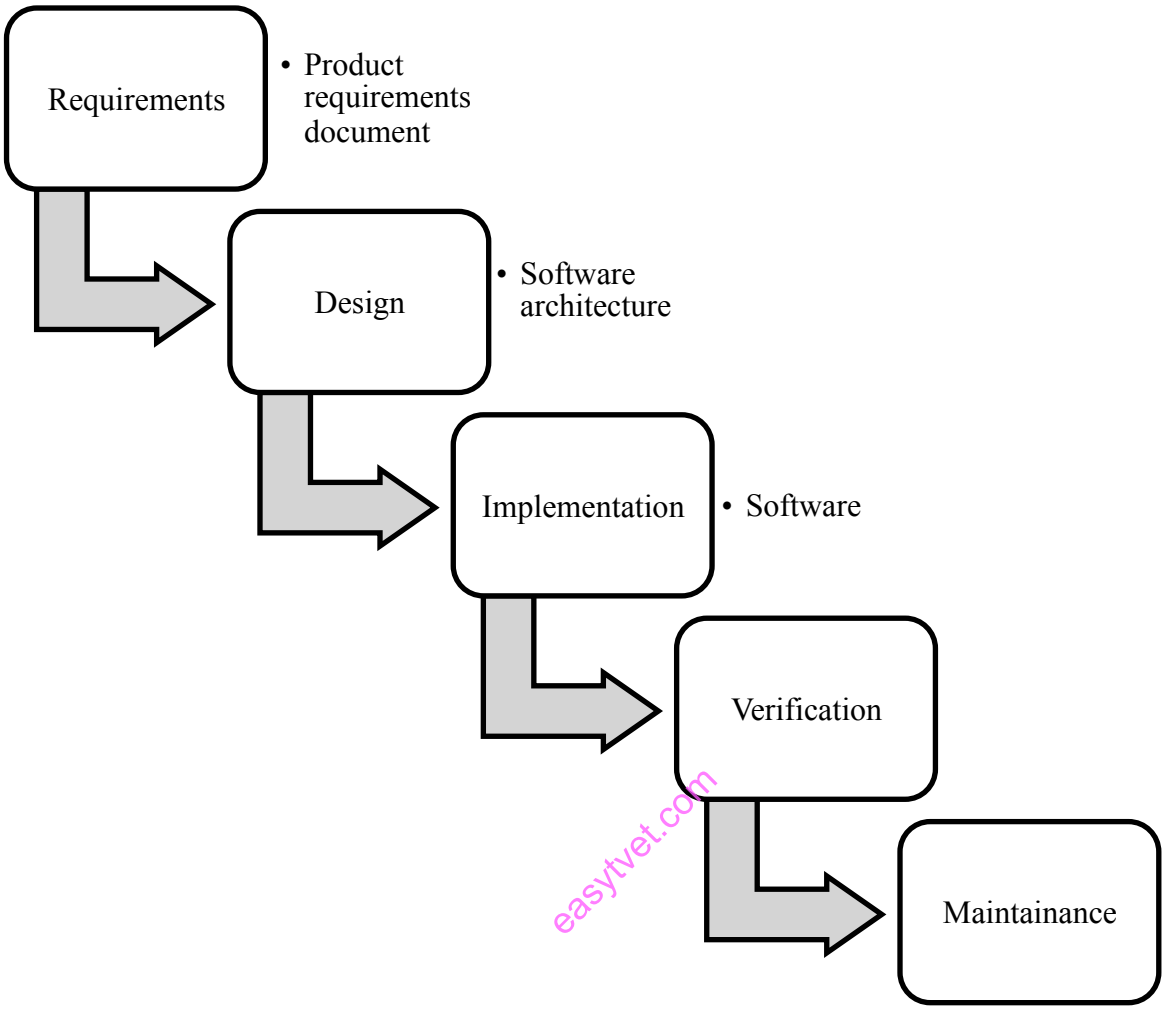
Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as attributes), and code, in the form of procedures (often known as methods).

A function is a block of organized, reusable code that is used to perform a single, related action.

Imperative programming is a programming paradigm that uses statements that change a program's state.

Declarative programming is a **programming** paradigm a style of building the structure and elements of computer programs that expresses the logic of a computation without describing its control flow.

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks.



Source: https://en.wikipedia.org/wiki/Waterfall_model

Figure 92: Waterfall model

Agile software development is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer/end user.

Spiral Model is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress.

Read: Introduction to computer programming:
http://cs.brown.edu/courses/cs173/2012/book/book.pdf

12.3.2.3 Self-Assessment

i. What is Agile software development model?
ii. What are the benefits of agile approach?
iii. What is waterfall approach?

iv. If you needed to execute some code repeatedly based on a certain condition, which of the following would you use?
   A. Variable
   B. Compiler
   C. Loop
   D. Conditional

v. Before source code can be compiled, it has to be _____
   A. Saved in a separate file
   B. Viewed in a command prompt
   C. Capitalized
   D. Parsed

vi. You need special software to write programs?
   A. True
   B. False

vii. What is object-oriented programming?
   A. A type of programming involving a structured method of creating programs
   B. A type of programming using only numbers
   C. A type of programming not in use anymore
   D. A type of programming involving data types representing data structures

12.3.2.4 Tools, Equipment, Supplies and Materials

Computer, Documentation, Flowcharts, Pseudocode, Programming language

12.3.2.5 References

- https://techterms.com/definition/programming_language
- https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/control-flow/loop-structures
- https://www.tutorialspoint.com/computer_programming/computer_programming_functions.htm
- Computer Programming: Learn It, Try It!, authored by Brad Edelman published by Capstone Global Library Ltd., 2017
- Computer Programming and Computer Systems authored by Anthony Hassitt, Anthony Ralston published by Academic Press, 2014

12.3.3 Learning Outcome 2: Identify phases of program development

12.3.3.1 Learning Activities

The following are the performance criteria:

- Processes of creating programs are identified
- Phases of program development are identified
- Activities that take place during Program Development are identified

Trainees to demonstrate knowledge in relation to:

- Phases of program development
- Planning
- System analysis and design
- System development
- Testing
- Implementation

12.3.3.2 Information Sheet

Requirement gathering and analysis or planning: Business requirements are gathered in this phase. This phase is the main focus of the project managers and stakeholders. Meetings with managers, stakeholders and users are held in order to determine the requirements like; Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

Design: In this phase the system and software design is prepared from the requirement specifications, which were studied, in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

Implementation: On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.

Testing: After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase all types of functional testing like unit testing, integration testing, system testing, acceptance testing are done as well as non-functional testing are also done.

Deployment: After successful testing the product is delivered / deployed to the customer for their use.

Read: Phase of program development: https://www.linkedin.com/pulse/what-software-development-life-cycle-sdlc-phases-private-limited

12.3.3.3 Self-Assessment

i. What are the phases of program development?
ii. Prepare a timeline based on the phases of development for a developing a website.
iii. Prepare a timeline based on the phases of development for a mobile application.
iv. Read about software development processes.
v. Risk analysis of a project is done in:
   A. System Analysis Phase
   B. Feasibility Study
   C. Implementation phase
   D. Maintenance phase
vi. Which is the first step in the software development life cycle?
   A. Analysis
   B. Design
   C. Problem/Opportunity Identification
   D. Development and Documentation
vii. Which tool is use for structured designing?
   A. Program flowchart
   B. Structure chart
   C. Data-flow diagram
   D. Module
viii. A step-by-step instruction used to solve a problem is known as:
   A. Sequential structure
   B. A List
   C. A plan
   D. An Algorithm
ix. An iterative process of system development in which requirements are converted to a working system that is continually revised through close work between an analyst and user is called
   A. Waterfall modeling
   B. Iterative modeling
   C. Spiral modeling
   D. None of these above

12.3.3.4 Tools, Equipment, Supplies and Materials

Computer, software

12.3.3.5 References

- https://www.linkedin.com/pulse/what-software-development-life-cycle-sdlc-phases-private-limited

- Computer Programming and Computer Systems authored by Anthony Hassitt, Anthony Ralston published by Academic Press, 2014
- Computer Programming: Learn It, Try It!, authored by Brad Edelman published by Capstone Global Library Ltd., 2017

12.3.4 Learning Outcome 3: Perform program design and analysis

12.3.4.1 Learning Activities

The following are the performance criteria:

- Program design and Analysis tools are identified
- Algorithm writing tools are identified
- Factors affecting program design and analysis are identified.
- System development methodologies are identified
- Criteria for choosing the appropriate methodology is done

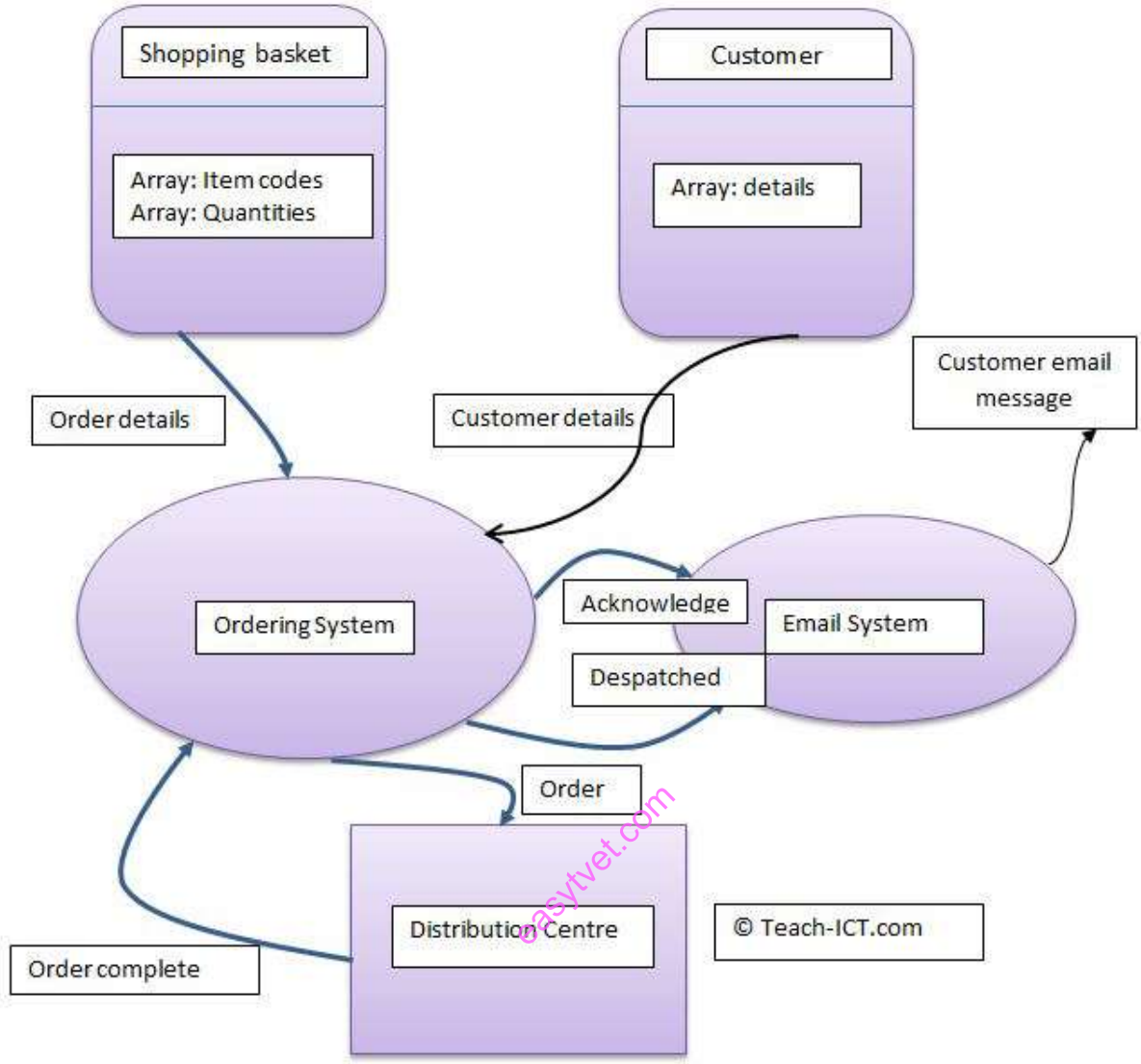
Trainees to demonstrate knowledge in relation to:

- Definition of program design and analysis
- Program design and analysis tools
- Dataflow diagram, Pseudocode, HIPO Diagram, Structure charts
- Software design levels
- High level design
- Detailed design
- Architectural design
- Types of system design
  Form design, File organization design, Database design

12.3.4.2 Information Sheet

Program design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. **Program analysis** helps the team understand key constraints that must be addressed for a program to be successful.

A data-flow diagram is a way of representing a flow of a data of a process or a system.



Source: https://www.teach-ict.com

Figure 93: Data flow

Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm.

HIPO (Hierarchical Input Process Output) diagram is a combination of two organized method to analyze the system and provide the means of documentation.

A **structure chart** in software engineering and organizational theory is a chart, which shows the breakdown of a system to its lowest manageable levels.

The architectural design is the highest abstract version of the system. It identifies the software as a system with many components interacting with each other. At this level, the designers get the idea of proposed solution domain.

The **high-level design** breaks the 'single entity-multiple component' concept of architectural design into less-abstracted view of sub-systems and modules and depicts their interaction with each other. High-level design focuses on how the system along with all of its components can be implemented in forms of modules. It recognizes modular structure of each sub-system and their relation and interaction among each other.

Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the previous two designs. It is more detailed towards modules and their implementations. It defines logical structure of each module and their interfaces to communicate with other modules.

Read: Types of system design:
https://www.tutorialspoint.com/system_analysis_and_design/system_design.htm

12.3.4.3 Self-Assessment

i. What is HIPO diagram?
ii. Define high-level design.
iii. Differentiate between data-flow diagram and HIPO diagram?
iv. What is pseudocode?
v. Explain software design levels?
vi. Which of the following data structure is not linear data structure?
   A. Arrays
   B. Linked lists
   C. Both of the above
   D. None of the above
vii. A system flow chart describes the:
   A. Details of each program module
   B. Line diagram for particular program
   C. Data files and operations and decision for a particular program
   D. Sequence of operations techniques is used to simplify defining problem
viii. Which of the following items are discussed during the system implementation phase of the application:
   A. Program Specification
   B. Software specification
   C. Software maintenance
   D. All of the above
ix. A system that groups a number of transactions for later processing is known is:
   A. Client server
   B. Batch system
   C. Online system
   D. Real time system

12.3.4.4 Tools, Equipment, Supplies and Materials

Lucidchart, Microsoft visio, Flow charts, Data flow diagram

12.3.4.5 References

- https://www.teach-ict.com/as_as_computing/ocr/H047/F451/312/slc_analysis/miniweb/pg4.htm
- https://www.tutorialspoint.com/software_engineering/software_design_basics.htm
- Computer Programming and Computer Systems, authored by Anthony Hassitt, Anthony Ralston published by Academic Press, 2014
- Computer Programming: Learn It, Try It!, authored by Brad Edelman published by Capstone Global Library Ltd, 2017

12.3.5 Learning Outcome 4: Develop a Computer program

12.3.5.1 Learning Activities

The following are the performance criteria:

- Format of a computer program is identified
- Fundamentals of structured programming using C language are done
- Fundamentals of Object Oriented programming using Java are done
- Well written and readable programs using disciplined coding styles and standards are adopted

Trainees to demonstrate knowledge in relation to:

- Format of a computer program
- Source code, Components of the program: Program header, declarations, main body Interrelationships between components, Data structures
- Fundamentals of structured programming using C language, Special features, Structure of C language, Variables and constants, Input/output functions, Literal reserved words, Identifiers, Data types and their sizes, Conditional statements, Loop control, C functions, Library functions, User defined functions, Arguments and parameters
- Fundamentals of Object Oriented programming using Java, Object oriented programming, Java language, Java Virtual Machine, Java Libraries, Program structure, Java Output, Variables and expressions, Classes and objects, Input in Java, Data types and operators, Boolean statements, Loops and program flow, Arrays, Exception handling

12.3.5.2 Information Sheet

Source code is the list of human-readable instructions that a programmer writes often in a word processing program when he is developing a program.

A data structure is a specialized format for organizing and storing data. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways.

C is called a structured programming language because to solve a large problem, C programming language divides the problem into smaller modules called functions or procedures each of which handles a particular responsibility.

A function is a group of statements that together perform a task. Every C program has at least one function, which is main, and all the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.

A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.

The C standard library provides numerous built-in functions that your program can call. For example, strcat() to concatenate two strings, memcpy() to copy one memory location to another location, and many more functions.

A function can also be referred as a method or a sub-routine or a procedure, etc.

A function definition in C programming consists of a function header and a function body. Here are all the parts of a function

Return Type: A function may return a value. The return_type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword void.

Function Name: This is the actual name of the function. The function name and the parameter list together constitute the function signature.

Parameters: A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.

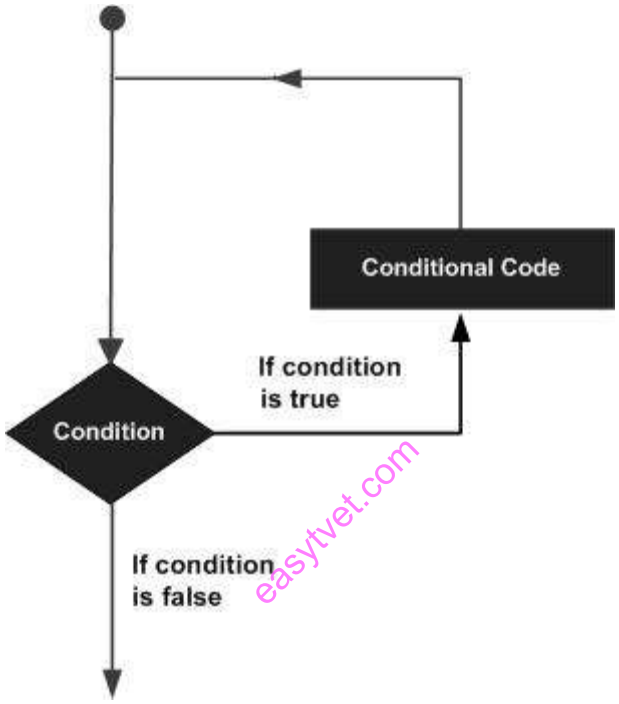
Function Body: The function body contains a collection of statements that define what the function does.

A function declaration tells the compiler about a function name and how to call the function. The actual body of the function can be defined separately.

Read: Programming in C: http://www.vssut.ac.in/lecture_notes/lecture1424354156.pdf

You may encounter situations, when a block of code needs to be executed several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.
Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times. Given below is the general form of a loop statement in most of the programming languages



Source: www.tutorialspoint.com

Figure 94: Loop statement

C programming language provides the following types of loops to handle looping requirements.

- **While loop:** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
- **For loop:** Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
- **Do...while loop:** It is more like a while statement, except that it tests the condition at the end of the loop body.
- **Nested loops:** You can use one or more loops inside any other while, for, or do..while loop.

A loop becomes **an infinite loop** if a condition never becomes false. The ‘for’ loop is traditionally used for this purpose. Since none of the three expressions that form the 'for' loop are required, you can make an endless loop by leaving the conditional expression empty.

```
#include <stdio.h>

  int main () {

    for( ; ; ) {

    printf("This loop will run forever.\n");    }

  return 0;

}
```

When the conditional expression is absent, it is assumed to be true. You may have an initialization and increment expression, but C programmers more commonly use the for(;;) construct to signify an infinite loop.

12.3.5.3 Self-Assessment

i. Explain the syntax for loop.
ii. Can a program be compiled without main() function?
iii. What is the advantage of declaring void pointers?
iv. What are command line arguments?
v. What is a variable?
vi. What is the output of the following program?


```
#include<stdio.h>

{

int x = 1;

switch(x)

 {

default: printf("Hello");

case 1: printf("hi"); break;

} }
```

A. Hello
B. Hi
C. HelloHi
D. Compile error

vii. What is JV? Why is Java called the “Platform Independent Programming Language”?
viii. What are the features in JAVA?
ix. What is a class and give an example of class?

x. If, a moderately active person cuts their calorie intake by 500 calories a day, they can typically lose about 4 pounds a month.
Write a Java and C program that lets the user enter their starting weight, then creates and displays a table showing what their expected weight will be at the end of each month for the next six month If they stay in this diet.

xi. Write a Java program that will calculate total of calories for every food classes (protein, carbohydrate and fat) and then total up the calories of all the food classes. The number of calories per gram of carbohydrate, fat and protein are 4, 7 and 9. The program will allow user to enter the number of grams for every food class.

xii. Write a C program to compute the sum of the two given integer values. If the two values are the same, then return triple their sum.

xiii. Which is a mechanism where one object acquires all the properties and behaviors of the parent object?
A. Inheritance
B. Encapsulation
C. Polymorphism
D. None of the above

xiv. Java Virtual Machine is platform independent.
A. True
B. False

xv. Who is father of C Language?
A. Bjarne Stroustrup
B. James A. Gosling
C. Dennis Ritchie
D. Dr. E.F. Codd

12.3.5.4 Tools, Equipment, Supplies and Materials

JDK (Java Development Kit), Eclipse IDE, Netbeans IDE, Junit, Codeblocks

12.3.5.5 References

- https://www.thoughtco.com/source-code-definition-958200
- http://www.vssut.ac.in/lecture_notes/lecture1424354156.pdf
- https://dzone.com/articles/15-tools-make-life-easy-java
- https://www.programiz.com/java-programming/basic-input-output
- https://www.tutorialspoint.com/cprogramming/c_functions.htm
- Java Programming 24-Hour Trainer (1st edition) authored by Yakov Fain published by Wrox; 2011
- Computer Programming and Computer Systems authored by Anthony Hassitt, Anthony Ralston published by Academic Press, 2014
- Computer Programming: Learn It, Try It!, authored by Brad Edelman published by Capstone Global Library Ltd., 2017

12.3.6 Learning Outcome 5: Perform Program testing and debugging

12.3.6.1 Learning Activities

The following are the performance criteria:

- Difference between testing and debugging is understood.
- Testing types, levels and methods are identified
- Debugging steps, requirements, principles and techniques are identified
- Error correction is done

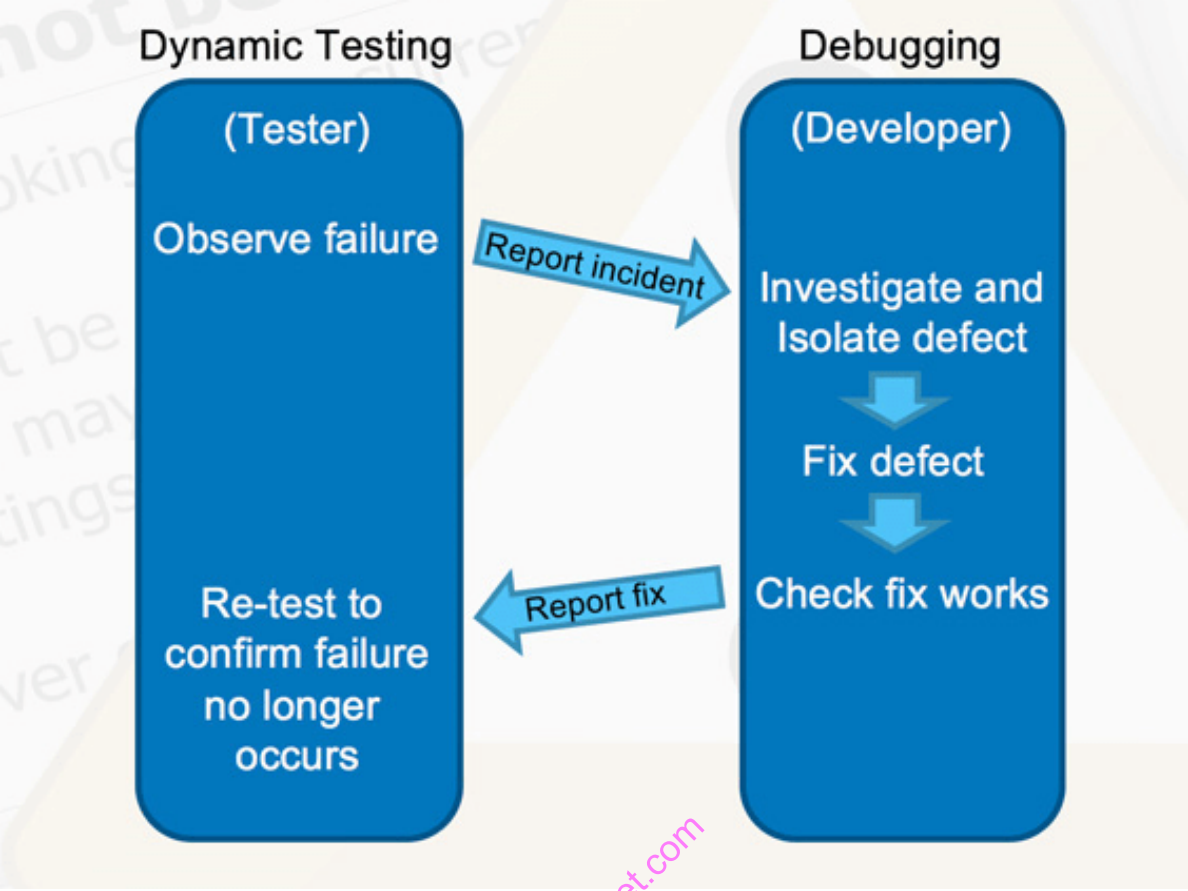
Trainees to demonstrate knowledge in relation to:

- Difference between testing and debugging.
- Types of testing: Smoke, Functional, Usability, Security, Performance, Regression, Compliance
- Levels of testing: Unit, Integration, System, Acceptance
- Methods of testing: Black box, White box, Gray box, Agile, Adhoc
- Debugging steps
- Debugging requirements
- Debugging principles
- Debugging techniques

12.3.6.2 Information Sheet

Testing: Fundamentally, testing is a process to check if the system is working same as it was supposed to do, and not working, as it was not supposed to do. It is done by the tester to identify the defects in the system (actual result of test case execution is not matching with expected result).

Debugging is the activity performed by developers to fix the bug found in the system. This is manual step-by-step unstructured and unreliable process to find and removes a specific bug from the system.



Source: https://dzone.com/articles/the-differences-between-testing-and-debugging

Figure 95: Dynamic testing and debugging

Smoke Testing: Whenever a new build is provided by the development team then the software testing team validates the build and ensures that no major issue exists.

Functional Testing: This type of testing ignores the internal parts and focuses only on the output to check if it is as per the requirement or not. It is a Black-box type testing geared to the functional requirements of an application.

Security Testing: It is a type of testing performed by a special team of testers. A system can be penetrated by any hacking way.

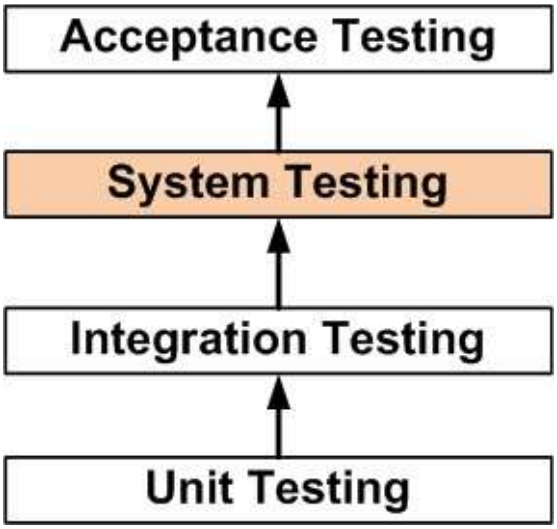
Usability Testing, user-friendliness check is done. Application flow is tested to know if a new user can understand the application easily or not. Proper help document is prepared if a user gets stuck at any point. Basically, system navigation is checked in this testing.

Performance Testing is done to check whether the system meets the performance requirements. Different performance and load tools are used to do this testing.

Regression Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the system in Regression Testing, so typically automation-testing tools are used for these types of testing.

Read: Software testing: https://www.softwaretestinghelp.com/types-of-software-testing/

Unit testing is a smallest testable portion of system or application, which can be compiled, liked, loaded, and executed. This kind of testing helps to test each module separately.



Source: guru99.com

Figure 96: Flow of unit testing

Read: Different type of testing: https://www.guru99.com/levels-of-testing.html

Black Box Testing: a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. Test design techniques include Equivalence partitioning, Boundary Value Analysis, Cause-Effect Graphing.

White Box Testing: a software testing method in which the tester knows the internal structure/design/implementation of the item being tested. Test design techniques include Control flow testing, Data flow testing, Branch testing, Path testing.

Gray Box Testing: A software testing method which is a combination of Black Box Testing method and White Box Testing method.

Agile Testing: A method of software testing that follows the principles of agile software development.

Ad Hoc Testing: A method of software testing without any planning and documentation.

Read: Debugging steps: https://kencoding.wordpress.com/2015/06/06/the-four-steps-of-debugging/

Watch: Software debugging steps: https://youtu.be/Kmx_NL4_2Fk

12.3.6.3 Self-Assessment

i. What is Functional Testing?
ii. Describe level of software testing?
iii. What is the difference between testing and debugging?
iv. Some incorrect word sequence in a program would generate
   A. Semantics error
   B. Syntax error
   C. Runtime error
   D. Logical error
v. Examination of the program step by step is called _____
   A. Controlling
   B. Tracing
   C. Stepping
   D. Testing
vi. The examination of changing values of variables is called stepping.
   A. True
   B. False
vii. _____ creates an inferior process that runs your program.
   A. run
   B. exit
   C. execute
   D. e

12.3.6.4 Tools, Equipment, Supplies and Materials

Test IO, TestCraft

12.3.6.5 References

- http://www.softwaretestingclass.com/difference-between-testing-and-debugging/
- https://www.guru99.com/levels-of-testing.html
- http://softwaretestingfundamentals.com/software-testing-methods/
- Computer Programming and Computer Systems authored by Anthony Hassitt, Anthony Ralston published by Academic Press, 2014
- Computer Programming: Learn It, Try It!, authored by Brad Edelman published by Capstone Global Library Ltd., 2017

12.3.7 Learning Outcome 6: Perform User training and Program Maintenance

12.3.7.1 Learning Activities

The following are the performance criteria:

- User training needs are identified
- Methods of user training are identified
- User training manuals are generated
- Maintenance schedule is developed
- Maintenance tools and techniques are determined.
- System performance is monitored, bugs are rectified and requested changes are made.

Trainees to demonstrate knowledge in relation to:

- Identification of user training needs
- Methods of user training
- User training manuals
- Maintenance schedule
- System maintenance tools and techniques.
- Monitoring of system performance
- Rectification of bugs
- Handling requested changes

12.3.7.2 Information Sheet

Read: User training method: https://www.hr.com/en/communities/training_and_development/list-of-training-methods_eacwezdm.html

Software maintenance in **software** engineering is the modification of a **software** product after delivery to correct faults, to improve performance or other attributes.

Read: Maintenance technics: https://www.researchgate.net/figure/Maintenance-tools-and-techniques_tbl1_227131529
http://ecomputernotes.com/software-engineering/tools-for-software-maintenance

Monitoring system performance is knowing whether a computer has issues is fairly straightforward when the computer is right in front of you. In order to keep your system fit for purpose, your monitoring activities need to cover the following priorities:

- Acceptable delivery speeds
- Constant availability
- Preventative maintenance
- Software version monitoring and patching
- Intrusion detection
- Data integrity
- Security monitoring
- Attack mitigation

- Virus prevention and detection

Rectification of bugs: A bug is any mismatch or an error in a program that produces unexpected results and prevents the software to behave correctly.

Read: How to handle request: https://www.researchgate.net/figure/Maintenance-tools-and-techniques_tbl1_227131529

12.3.7.3 Self-Assessment

i. What is software maintenance?

ii. Maintenance consist of the following action(s)
   A. Replace of component
   B. Repair of component
   C. Service of component
   D. All of the above

iii. Software Maintenance includes
   A. Error corrections
   B. Enhancements of capabilities
   C. Deletion of obsolete capabilities
   D. All of the mentioned

iv. The modification of the software to match changes in the ever changing environment, falls under which category of software maintenance?
   A. Corrective
   B. Adaptive
   C. Perfective
   D. Preventive

v. _____ measures the ability of a regression test selection technique to handle realistic applications.
   A. Efficiency
   B. Precision
   C. Generality
   D. Inclusiveness

12.3.7.4 Tools, Equipment, Supplies and Materials

Debugger, ManageEngine OpManager, SolarWinds Server Health Monitor

12.3.7.5 References

- https://www.researchgate.net/figure/Maintenance-tools-and-techniques_tbl1_227131529
- http://ecomputernotes.com/software-engineering/tools-for-software-maintenance
- https://www.hr.com/en/communities/training_and_development/list-of-training-methods_eacwezdm.html
- https://www.researchgate.net/figure/Maintenance-tools-and-techniques_tbl1_227131529

- Computer Programming and Computer Systems authored by Anthony Hassitt, Anthony Ralston published by Academic Press, 2014
- Computer Programming: Learn It, Try It!, authored by Brad Edelman published by Capstone Global Library Ltd., 2017