17.2.3. OBJECT ORIENTED PROGRAMMING (90 HOURS)

17.2.3.01 INTRODUCTION

This module unit is intended to provide the trainee with knowledge and skills to develop programs in Object Oriented Languages.

17.2.3.02 GENERAL OBJECTIVES

By the end of this module unit the trainee should be able to:

a) understand the various data types, control structures and data structures used in object oriented programming

b) apply programming skills in C++

c) develop object oriented programs

17.2.3.03 COURSE SUMMARY AND TIME ALLOCATION

NB: APPROPRIATE TEACHING LANGUAGES - C++

CODE	TOPIC	SUBTOPIC	HOURS T	HOURS P	TOTAL
17.2.3.1	INTRODUCTION TO OBJECT ORIENTED PROGRAMMING	• object oriented programming • evolution of object oriented programming • OOP paradigms • merits and demerits of OOP • examples of object oriented languages • operating systems requirements • object oriented databases (OODBs)	4		4
17.2.3.2	OOP CONCEPTS	• concepts associated with OOP • comparison between structured and OOP • reasons for embracing OOP	8		8

CODE	TOPIC	SUBTOPIC	HOURS T	HOURS P	TOTAL
17.2.3.3	LANGUAGE STRUCTURES OF OBJECT ORIENTED PROGRAMMING (OOP)	• language structure • Features of OOP languages • File extensions in OOP • data types in OOP • variable declaration • implementation of language structure	4	12	16
17.2.3.4	ESSENCE OF OBJECTS AND CLASSES	• definition of objects and classes in OOP • importance of objects and classes in OOP • implementation of objects and classes	6	13	19
17.2.3.5	INHERITANCE	• meaning and importance • rules of inheritance in OOP • types of inheritance in OOP • implementation of inheritance	2	6	8
17.2.3.6	POLYMORPHISM	• meaning and importance of polymorphism • encapsulation/information Hiding • implementation of polymorphism	2	6	8
17.2.3.7	CONSTRUCTORS AND DESTRUCTORS	• meaning of constructors • constructor implementation	2	8	10
17.2.3.8	OPERATOR OVERLOADING	• meaning and importance of operator overloading • implementation of operator overloading	2	8	10
17.2.3.9	FILE ORGANISATION	• meaning and importance of file organization • file stream • file stream features/properties • file operations	2	2	4

CODE	TOPIC	SUBTOPIC	HOURS T	HOURS P	TOTAL
17.2.3.10	EMERGING TRENDS IN OBJECT ORIENTED PROGRAMMING	• emerging trends in OOP • challenges of emerging trends in OOP • coping with challenges of emerging trends in OOP	2	2	2

17.2.3.1T INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

THEORY

17.2.3.1.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) define Object Oriented Programming
b) trace the evolution of Object Oriented Programming
c) describe the different programming paradigms
d) explain the merits and demerits of OOP
e) describe operating systems requirements for OOP
f) identify example of Object Oriented Programming languages
g) describe Object Oriented Databases (OODBs)

CONTENT

17.2.3.1.T1 Definition of Object Oriented Programming

17.2.3.1.T2 Evolution of object oriented programming

17.2.3.1.T3 Programming paradigms

- unstructured programming
- procedural programming
- modular programming
- object oriented programming

17.2.3.1.T4 Merits and demerits of OOP

17.2.3.1.T5 Operating system requirements

17.2.3.1.T6 Examples of object oriented languages

- C++
- java
- others

17.2.3.1.T7 Object Oriented Databases (OODBs)

hybrid object oriented databases

persistent object oriented databases

pure object oriented databases

17.2.3.2T OBJECT ORIENTED PROGRAMMING CONCEPTS

THEORY

17.2.3.2.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) explain OOP concepts

b) compare structured Vs OOP

c) explain the reasons for embracing OOP

CONTENT

17.2.3.2.T1 Concepts associated with OOP

class

object

relationship

inheritance

polymorphism

encapsulation

17.2.3.2.T2 Comparison between structured and OOP

keywords and identifiers

comments

literals

constants

punctuators

17.2.3.2.T3 Reasons for embracing OOP

17.2.3.3T LANGUAGE STRUCTURES OF OBJECT ORIENTED PROGRAMMING (OOP)

17.2.3.3.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) describe language structure

b) describe the features of OOP languages

c) identify file extensions in OOP

d) describe data types in OOP
e) describe variable declarations

CONTENT

17.2.3.3.T1 Language structure

17.2.3.3.T2 Features of OOP languages

17.2.3.3.T3 File extensions in OOP

17.2.3.3.T4 Data Types in OOP

simple data type

derived types

- pointers
- reference
- arrays
- structures
- functions
- class types

17.2.3.3.T5 Variable declaration

declaration syntax

initialization

data conversion

scope of variables

type conversion

explicit type checking

PRACTICE

17.2.3.3.P0 **Specific Objectives**

By the end of this topic, the trainee should be able to:

a) implement language structure in C+++

17.2.3.3.P1 Implementing language structure in C+++

declarations

operators

extensions

statements

CONTENT

17.2.3.3.P2 Implement language structure in specific OOP language

declarations

operators

extensions

statements

17.2.3.4T ESSENCE OF OBJECTS AND CLASSES

THEORY

17.2.3.4.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) explain objects and classes in OOP

b) explain importance of objects and classes in OOP

CONTENT

17.2.3.4.T1 Objects and classes in OOP

17.2.3.4.T2 Importance of objects and classes in OOP

17.2.3.4.T3 Implementation of objects and classes

initialization
free store
static objects
implicit pointer
in-line function
friend of class
static class members
specifiers – const, enum, typedef
enumerated constant
pointer to members
nested classes
container class libraries

PRACTICE

17.2.3.4.P0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) implement object and classes using C+++

CONTENT

17.2.3.4.P1 Implementation of objects and objects using C+++

initialization
free store
static objects
implicit pointer
in-hire function etc

17.2.3.5T INHERITANCE

THEORY

17.2.3.5.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) explain the importance of inheritance
b) describe the rules of inheritance in OOP
c) describe the types of inheritance in OOP
d) implement inheritance

CONTENT

17.2.3.5.T1 Meaning and importance of inheritance

17.2.3.5.T2 Rules of inheritance in OOP

17.2.3.5.T3 Types of inheritance in OOP

PRACTICE

17.2.3.5.P0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) implement inheritance

CONTENT

17.2.3.5.P1 Implementation of inheritance

- derived classes
- inheritance and friends
- pointers to objects
- inheritance and constructors
- inheritance and destructors
- order of constructor invocation
- multiple inheritance
- base class conversions
- standard conversions
- user-defined conversions
- inheritance and class scope
- inheritance and overloading
- inheritance relationship

17.2.3.6T POLYMORPHISM

THEORY

17.2.3.6.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) explain meaning and importance of polymorphism

b) explain encapsulation / information hiding

CONTENT

17.2.3.6.T1 Meaning importance of polymorphism

17.2.3.6.T2 Encapsulation / Information hiding

virtual functions & abstract classes

ambiguity

virtual base class

PRACTICE

17.2.3.6.P0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) implement polymorphism

CONTENT

17.2.3.6.P1 Implement polymorphism

virtual functions & abstract classes

ambiguity

virtual base class

17.2.3.7T CONSTRUCTORS AND DESTRUCTORS

THEORY

17.2.3.7.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) explain the meaning and importance of constructors and destructors

CONTENT

17.2.3.7.T1 Meaning and importance of constructors and destructors

default constructors

copy constructors

argument matching

destructors

overloading& scope

PRACTICE

17.2.3.7.P0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) implement constructors

17.2.3.7.P1 Implementation of constructors

- default constructors
- copy constructors
- argument matching
- overloading& scope

17.2.3.7.P2 Implementation of destructors

17.2.3.8T OPERATOR OVERLOADING

THEORY

17.2.3.8.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) meaning and importance of operator overloading

CONTENT

17.2.3.8.T1 Meaning and importance of operator overloading

PRACTICE

17.2.3.8.P0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) a) implement operator overloading

CONTENT

17.2.3.8.P1 Implement operator overloading

- expression
- binary and unary operators
- operator function
- over loadable operators
- rules
- predefined meaning for operators
- operators new and delete
- conversion operators
- ambiguities
- subscripting, function call & dereferencing

- subscript
- function call
- dereferencing
- friend operators

17.2.3.9T FILE ORGANISATION

THEORY

17.2.3.9.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) explain the meaning and importance of file organization
b) describe file stream
c) describe file stream features / properties
d) implement file operation

CONTENT

17.2.3.9.T1 Description of file organization

file input / output

17.2.3.9.T2 Description of File Stream

file stream Input/Output

17.2.3.9.T3 File Stream features/properties

stream class hierarchy
reference
member functions
istream class
ostream class

PRACTICE

17.2.3.9.P0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) implement file operations

CONTENT

17.2.3.9.P1 Implementing file operations

record appending
record insertion
record modification
record deletion

17.2.3.10T EMERGING TRENDS IN OBJECT ORIENTED PROGRAMMING

THEORY

17.2.3.10.T0 Specific Objectives

By the end of this topic, the trainee should be able to:

a) identify emerging trends in OOP
b) explain the challenges of emerging trends in OOP
c) cope with the challenges of emerging trends in OOP

CONTENT

17.2.3.10.T1 Emerging trends in OOP

17.2.3.10.T2 Challenges of emerging trends in OOP

17.2.3.10.T3 Coping with challenges in OOP

TEACHING/LEARNING RESOURCES

Relevant text books and free e-books
www contents
Sample codes from www contents
OOP languages

ASSESSMENT MODE

Written tests
Practical tests
Program project development using OOP concepts in programming
Oral tests