**COURSE CODE: HCS124/ISH21**

**COURSE TITLE: OBJECT ORIENTE PROGRAMMING OOP**

**LECTURER: MS M. MADYEMBWA**

**CELL: 0771151011**

**EMAIL: mmadyembwa@gzu.ac.zw**

**Course outline Lecture 1:** **Introduction to Core Programming Concepts**

This Lecture provides background and foundational information on how computers process information, discusses the different types of applications that a programmer might be creating, and then provides information on how code is compiled and interpreted by a computer.

**Lecture 2:** **Core Programming Language Concepts**

This Lecture covers programming language syntax and the importance of using good syntax and following the syntax rules for the chosen language. This Lecture also discusses the core data types and how to store these data types in computer memory by using variables and constants

**Lecture 3**: **Program Flow**

This Lecture covers how code is executed in a computer program, such as top to bottom, in structured programming and branching in code execution. The Lecture teaches these concepts using functions, decision structures, and looping construct

**Lecture 4**: **Algorithms and Data Structures**

This Lecture introduces the concept of an algorithm by outlining all the steps required including the decisions to be made as the routine progresses. The Lecture also discusses how to translate these set of steps into pseudocode for evaluation of the algorithm that will be translated into actual code

**Lecture 5**: **Error Handling and Debugging**

This Lecture helps students understand that errors are a part of programming, and they must understand how to anticipate errors, handle those errors in code, and present a good user experience. This Lecture introduces structured exception handling as a mechanism to deal with errors.

**Lecture 6**: **Introduction to Object-Oriented Programming**

This Lecture covers an introduction to the concepts related to object-oriented programming (OOP). The content has been split across two Lectures with this Lecture focusing on basic OOP concepts that will provide sufficient knowledge to understand complex data structures starting with structs and then moving on to classes. This Lecture helps the students gain an understanding of how to encapsulate data and related functionality within a class.

**Lecture 7**: **More Object-Oriented Programming**

This Lecture teaches students about inheritance and polymorphism in classes and function overloading. Function overloading and polymorphism often go hand-in-hand, such as when you inherit from a class, or when you want to override or change the existing behaviour to suit the needs of your class

**Lecture 8: Introduction to Application Security and Network Programming:**

This Lecture helps students think about security in their applications. It introduces the concepts of authentication and authorization for users, and also introduces the concept of permissions for running code. It explains that operating systems might prevent certain aspects of the program from executing, such as saving a file to a directory to which the user running the app might not have permission to write. The Lecture briefly covers code signing and why programmers might want to consider using it.

**Lecture 9**: **Core I/O Programming**

This Lecture introduces some core input/output (I/O) concepts that programmers will use while creating applications. Starting with console I/O, this Lecture introduces input and output to the Console window. The Lecture also talks about reading from and writing to the filesystem.

**Lecture 10**: **Application Performance and Memory Management**

This Lecture enables students to understand that memory on a computer is a finite resource. It talks about how good application design and good coding discipline with memory management will help programmers learn to develop applications that are fast, responsive, and do not negatively impact other applications.

**Assignments and Test:**

|  |  |
| --- | --- |
| **Work** | **Dates** |
| 2 practical assignments | TBA |
| 1 Inclass Test | TBA |

**Books**

* Thinking in Java
* Any oracle java certification textbooks

**IDE**

**Netbeans**