|  |  |  |
| --- | --- | --- |
| **Course-17 Title: Electronic Devices and Circuits** | |  |
| Course No.: EEE 122 Credit : 1.5 Contact Hours: 3 | Total Marks: 100 | |

**11.1 Rationale:** In order to serve the nation as a computer engineer its practitioner needs to apply basics of electronics in electronic applications.

* 1. **Objectives:**

1. To familiarize with different electronic components and measuring instruments.
2. To design and performance analysis of different electronic circuits.

|  |  |  |  |
| --- | --- | --- | --- |
| **11.3**  **Learning Outcomes** | **11.4**  **Course Content** | **11.5**  **Teaching Strategy/ Learning Experience** | **11.6 Assessment Strategy** |
| 1. Identify electronic components. 2. Demonstration of measuring instruments | Familiarization with different electronic devices and measuring instruments. | Lecture, Demonstration | Short answer, Assignment, Viva, reports |
| 1. Design and analyze performance of different electronic circuits. | Diode, Transistor, UJT,SCR, Diac, Traic,Zener diode, LED | Lecture, Demonstration, Group Discussion | Short answer, Assignment, Viva, reports |

**RECOMMENDED BOOKS AND PERIODICALS**

**Recommended Books**:

1. V.K. Mehta : Principles of Electronics
2. R.L. Boylestad : Electronic Devices and Circuit Theory
3. Millman & Halkias: Electronic Devices and Circuits

|  |  |
| --- | --- |
| **Course-18 Title: Objective Oriented Programming** |  |
| **Course No.:CCE-121 Credit : 3 Contact Hours: 3** | **Total Marks: 100** |

**11.1 Rationale:**

This course is designed to teach Object-Oriented programming concepts, techniques, and applications using the Java programming language.

**11.2 Objectives:**

1. The **model** of object oriented programming: **abstract data types, encapsulation, inheritance** and **polymorphism**
2. **Fundamental features** of an object oriented language like Java: **object classes** and **interfaces, exceptions** and libraries of **object collections**
3. How to take the statement of a business problem and from this determine **suitable logic for solving the problem;** then be able to proceed to code that logic as a program written in Java.
4. How to **test, document** and **prepare** a professional looking package for each business project using javadoc.

|  |  |  |  |
| --- | --- | --- | --- |
| **11.3**  **Learning Outcomes** | **11.4**  **Course Content** | **11.5**  **Teaching Strategy/ Learning Experience** | **11.6 Assessment Strategy** |
| * Explain advantage of OOP * Differentiate between OOP and structure language | Philosophy of Object Oriented Programming (OOP)  Advantages of OOP over structured programming | **Lecture** | **Essay question** |
| * Create class and object * Explain encapsulation * Distinguish access modifier * Define static and no static member | Concepts of classes and objects, Encapsulation, access specifiers, static and non-static members. | **Lecture, Exercise** | **Short question, multiple choice, assignment** |
| * Explain control structure * Apply event handling in software development. | JAVA applications; event handling; control structures. | **Lecture, Exercise** | **Assignment,**  **Essay question** |
| * Differentiate constructor and destructor * Apply single and multidimensional array in development * define overloaded | Methods; Overloaded Methods, Constructors, destructors ,Single Multi-Dimension Arrays | **Lecture,Exercise** | **Assignment,quiz,short question.** |
| * Discuss interface in respect of OOP | Object Based Programming Object oriented Programming interface. | **Lecture,exercise** | **Assignment** |
| * Explain heritance * Create inheritance based program * Apply inheritance in development | Inheritance: single and multiple inheritance | **Lecture,exercise** | **Assignment,**  **quiz,short question.** |
| * Diferentiate between method overload and override * Contrast between abstract class and interface | Polymorphism: overloading, abstract classes, virtual functions and overriding | **Lecture,exercise** | **Assignment,quiz,short question.** |
| * Define exception * Discuss different types of exception * Create custom exception | Exceptions Handling | **Lecture,exercise** | **Assignment,quiz,short question.** |
| * Define string * Manipulate string operation * Create string processing algorithm | String manipulation | **Lecture,exercise** | **Assignment,quiz,short question.** |
| * Create GUI * Apply mouse and keyboard event | Introduction to graphical user interface; handling mouse and keyboard events | **Lecture,exercise** | **Assignment,quiz,short question.** |
| * Design multithread program * design client server program * define thread | Multi Threading;Client Server programming. | **Lecture,exercise** | **Assignment,quiz,short question.** |

**RECOMMENDED BOOKS AND PERIODICALS**

1. Java™ How to Program, Latest Edition **by** H. M. Deitel -  Deitel & Associates, Inc., P. J. Deitel -  Deitel & Associates, Inc

2. Liang, Y. D. (2005), Introduction to Java Programming, Comprehensive: International Edition, 5/E, Prentice Hall. ISBN 0-13-185721

Additional Useful Texts

Barnes, D. & Kolling, M. 2005, Objects First with Java: A Practical Introduction using BlueJ, 2nd edn, Pearson.

Farrell, J. 2005, Programming Logic and Design, Comprehensive, 3rd edn, Thomson.

Malik, D. & Nair, P. 2003, Java Programming: From Problem Analysis to Program Design, Thomson.

Robertson, L. 2003, Simple Program Design: A Step-by-Step Approach, 4th edn, Thomson.

Russell, J. 2001, Java Programming for the Absolute Beginner, Premier Press. Shelly, G., Cashman, T., Starks, J. & Mick, M. 2004, Java Programming Comprehensive Concepts and Techniques, Thomson.

Sun Java technology: http://java.sun.com/

Borland JBuilder: http://www.borland.com/jbuilder

Web page for the prescribed text:

<http://www.cs.armstrong.edu/liang/intro5e/intro5estudentsolution.html>