**MANIPAL UNIVERSITY JAIPUR**School of Computing and Information Technology

**DEPARTMENT OF COMPUTER AND COMMUNICATION ENGINEERING**   
Course Hand-out

Object Oriented Programming using Java | CC 2104 | 4 Credits | 3 1 0 4

Session: Aug 20-Dec 20 | Faculty: Dr. Punit Gupta, Dr. Sourabh Singh Verma and Dr. Ghanshyam Raghuwanshi | Class: BTech CCE III SEM | Sec: A|B|C

**MANIPAL UNIVERSITY JAIPUR**School of Humanities and Social Sciences

**DEPARTMENT OF LANGUAGES**   
Course Hand-out

Course Name | Course Code| 4 Credits | 3 1 0 4

Session: Jan 20-May 20 | Faculty: Your Name| Class: BA ENGLISH (HONS) II/IV/VI SEM



1. **Introduction:** This course is offered by the Department of Computer and Communication Engineering as object oriented techniques have revolutionized the software development process and are used tremendously in IT industry to develop software products of various kinds. The course is designed to give students an in-depth understanding of the basic concepts of object-oriented programming such as encapsulation, inheritance and polymorphism using Java programming language as an aid in tool. The course curriculum and structure has been divided into eight basic modules which covers the programming aspects related with object oriented domain such as exception handling, multithreading, GUI programming, event handling etc. The course will be taught with the help of several teaching aides such as power point presentation and via live debugging and execution demonstrations of several programming problems using Eclipse tool. The main objective of the course is to teach students about the basics of classes and objects using Java programming language, to enable the students to properly use the basic object oriented pillars such as encapsulation, inheritance and polymorphism, to enable the students to understand the basic difference between a class and an interface, to teach students about the implementation aspect of various basic data structures such as Linked Lists and Arrays using object oriented techniques.
2. **Course Outcomes:** At the end of the course, students will be able to:
3. Apply to compile and execute Java Application using Command Based Interface as well as using Eclipse Tool.
4. Identify and implement the concepts of encapsulation and abstraction using class, objects and interfaces for better programming skills.
5. Describe and Implement various inheritance and polymorphism forms using Java Classes and Interfaces.
6. Implement various collection data structure such as linked lists, queues, stacks using Java’s collection framework
7. Apply, Learn and finally implement the use of advanced programming constructs/features such as exception handling, multithreading and event handling in real-life programming domains.
8. Visualize a real world problem in the form of various collaborating classes and objects for enhancing employability.
9. **PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES**

**PO**1. **Engineering knowledge**: Apply the knowledge of mathematics, computer science, and communication engineering fundamentals to the solution of complex engineering problems.

**PO**2. **Problem analysis**: The sophisticated curriculum would enable a graduate to identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using basic principles of mathematics, computing techniques and communication engineering principles.

**PO**3. **Design/development of solutions**: Upon analysing, the B Tech CCE graduate should be able to devise solutions for complex engineering problems and design system components or processes that meet the specified requirements with appropriate consideration for law, safety, cultural & societal obligations with environmental considerations.

**PO**4. **Conduct investigations of complex problems**: To imbibe the inquisitive practices to have thrust for innovation and excellence that leads to use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO**5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

**PO**6. **The engineer and society**: The engineers are called society builders and transformers. B. Tech CCE graduate should be able to apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO**7. **Environment and sustainability**: The zero effect and zero defect is not just a slogan, it is to be practised in each action. Thus a B Tech CCE should understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO**8. **Ethics**: Protection of IPR, staying away from plagiarism are important. Student should be able to apply ethical principles and commit to professional ethics, responsibilities and norms of the engineering practice.

**PO**9. **Individual and team work**: United we grow, divided we fall is a culture at MUJ. Thus an outgoing student should be able to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO**10. **Communication**: Communicate effectively for all engineering processes & activities with the peer engineering team, community and with society at large. Clarity of thoughts, being able to comprehend and formulate effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO**11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in varied environments.

**PO12. Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**[PSO.1] Clearly imbibe the basic principles, concepts and applications of computer based Communication/networking, information sharing, signal processing, web based systems, smart devices and communication technology.**

**[PSO.2] Investigate problematic areas prevalent in the field of Computer and Communication Engineering to find acceptable solutions.**

**[PSO.3] Identify the existing open problems in the field of computing and propose the best possible solutions.**

**[PSO.4] Apply the contextual knowledge in the field of computing and communication to assess social, health, safety and cultural issues and endure the consequent responsibilities relevant to the professional engineering practice.**

1. **Assessment Plan:**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Description** | **Maximum Marks** |
| Internal Assessment (Summative) | Sessional Exam I (Closed Book) | 20 |
| Sessional Exam II (Closed Book) | 20 |
| In class Quizzes and Assignments , Activity feedbacks (Accumulated and Averaged) | 20 |
| End Term Exam  (Summative) | End Term Exam (Closed Book) | 40 |
|  | Total | 100 |
| Attendance  (Formative) | A minimum of 75% Attendance is required to be maintained by a student to be qualified for taking up the End Semester examination. The allowance of 25% includes all types of leaves including medical leaves. | |
| Make up Assignments  (Formative) | Students who misses a class will have to report to the teacher about the absence. A makeup assignment on the topic taught on the day of absence will be given which has to be submitted within a week from the date of absence. No extensions will be given on this. The attendance for that particular day of absence will be marked blank, so that the student is not accounted for absence. These assignments are limited to a maximum of 5 throughout the entire semester. | |
| Homework/ Home Assignment/ Activity Assignment  (Formative) | There are situations where a student may have to work at home, especially before a flipped classroom. Although these works are not graded with marks. However, a student is expected to participate and perform these assignments with full zeal since the activity/ flipped classroom participation by a student will be assessed and marks will be awarded. | |

1. **SYLLABUS**

**Introduction:** OOP paradigm, the creation of java, the java buzzwords; C, C++ and Java comparison;**Java Basics:** Compilation and execution of a java program, access modifiers, garbage collection; **Class and Objects:** Class definition, creating objects, role of this keyword, garbage collection, finalize() method, method overloading, objects as parameters, argument passing, returning objects, access control, final, nested and inner classes; **I/O Basics:** Reading console input, writing console output, Files **Array and Strings:** Arrays in java, 1-D, 2-D and dynamic arrays, string basics, string comparison and manipulation; **Inheritance:** Inheritance and its types,abstract class, inner and outer class, super, final, static keywords; **Package and Interface:** In-built packages and user define packages, role of interface, polymorphism via inheritance; **Collection Framework & Generics:** List, set, map, generic classes; **Exception Handling:** Errors and exceptions, types of exceptions, handling exceptions, **Multithreading:** Thread class, runnable, thread life cycle, synchronization, thread priority; **Event Ha ndling and GUI Programming:** Events, action listener, swing package; /

1. **REFERENCE BOOKS** 
   1. Java:The Complete Reference (9th Edition), By Herbert Schildt, McGraw Hill Education.
   2. Balagurusamy E, ”Object Oriented Programming with Java”, Tata McGraw Hill, 2011.
   3. Arnold K, & Gosling J, “The Java Programming Language”, 2002.
   4. Horstmann CS, “Big Java”, Wiley’s Interactive Edition, 2015.
2. **Lecture Plan:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class  Number | Topics | Session Outcome | Mode of Delivery | Corresponding Course Outcome | Mode of Assessing the Outcome |
| 1 | Introduction and Course Hand-out briefing | To acquaint and clear teachers expectations and understand student expectations | Lecture | NA | NA |
| 2 | Overview of Java: History, Evolution, C, C++ and Java Comparison, Java Byte Code ,Java Buzzwords, Java SE 8 | Identify and implement the concept of OOP Java | Lecture | CO 1 | Mid Term I, Quiz & End Term |
| 3-7 | Lexical issues, java keywords, OOP Programming, Control Statements, Operators | Describe the programming constructs of OOP Java | Lecture | CO 1 | Mid Term I, Quiz & End Term |
| 8-11 | Data Types, Variables and Arrays: Primitive Types, Floating point, Characters, Booleans, Literals, Variables, Type Conversion and casting, wrapper classes, Boxing and Unboxing, 1D Arrays, 2D Array, multi dimension Array, Variable Length Array | Recall programming construct of OOP Java | Flipped Class | CO 1 | Mid Term I, Quiz & End Term |
| 12-17 | Introduction to classes: Class Fundamentals, Declaring Objects Methods in Classes, returning values, parameterized methods Garbage Collection, finalize() method Constructors, parameterized constructors This keyword, This Constructor, Constructor Chaining | Identify and implement the concepts of class and objects. | Lecture | CO 2 &  CO 6 | Mid Term I, Quiz & End Term |
| 18-23 | Classes and its Methods: Overloading Methods, Using Objects as parameters, Argument passing, Returning Objects, Recursion, Access Control Static, final, Nested and Inner class, Variable length arguments | Apply and describe the concept of interaction for classes and its methods | Lecture | CO 2 | Mid Term I, Quiz & End Term |
| 24-26 | I/O Basics: Using Command line arguments ,I/O Basics, reading Console Input and Writing Console Output, PrintWriter Class, Scanner Class, reading and Writing Files, Closing files | Recall I/O basics of OOP Java | Flipped Class | CO 1 & CO 2 | Mid Term II, Quiz & End Term |
| 27-29 | Inheritance: Basics, Using Super, Creating multilevel hierarchy, Method overriding, Dynamic method dispatch, Using Abstract class, using final with Inheritance  Tutorial | Examine and describe the concept of Inheritance of OOP Paradigm | Lecture | CO 3 | Mid Term II, Quiz & End Term |
| 30-34 | Packages, Access protection, Importing packages, static import Interfaces: Default interface methods, static methods in interfaces | Experiment the access control using package and interfaces | Lecture, activity | CO 3 | Mid Term II, Quiz & End Term |
| 35-38 | Exception Handling: Fundamentals, Exception types, Uncaught Exceptions, check uncheck Exception, Using try and catch, multiple catch clauses, nested try statements, Throw, throws, finally, built-in exceptions, creating own exception Sub classes | Recall and implement exceptions in classes | Lecture | CO 2, CO 5 & CO 6 | Mid Term II, Quiz & End Term |
| 39-42 | Multithreaded Programming: Thread Model: thread priorities, synchronization, main thread, creating single thread and multiple threads, using isAlive(), join(), Interthread communication, suspending, resuming and stopping threads, using multithreads | Implement and describe the concept of multithreading | Flipped Class | CO 2 & CO 6 | Quiz & End Term |
| 43-45 | String Handling: Constructors, Constructor chaining, string operations, Character extraction, comparison, searching and modifying, String Class Methods and String Buffer Class | Recall and experiment string handling in OOP Java | Lecture | CO 4 | Quiz & End Term |
| 46-49 | Generic Class: Collection framework, Arraylist , LinkList, HashMap,Vector, Making own generics class | Implement and describe the generic class and collection framework | Lecture | CO 4 | Quiz & End Term |
| 50-52 | GUI and Event Handling: GUI lifecycle, Events, Events listener, adapter classes, Different Event classes, Event Listener Interfaces | Apply GUI and Event handling using adapter class and interfaces | Flipped Class | CO 4 & CO 5 | Quiz & End Term |

1. **Course Articulation Matrix: (Mapping of COs with POs & PSOs)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **STATEMENT** | Correlation with Program Outcomes(POs) | | | | | | | | | | | | Correlation with Program Specific Outcomes (**PSOs**) | | | |
| PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 | PSO 3 | PSO 4 |
| [CC 2104.1] | Apply to compile and execute Java Application using Command Based Interface as well as using Eclipse Tool. | **1** | **2** | **2** | **2** | **-** | **-** | **-** | **-** | **1** | **1** | **1** | **1** | **2** | **-** | **-** | **-** |
| [CC 2104.2] | Identify and implement the concepts of encapsulation and abstraction using class, objects and interfaces for better programming skills. | **2** | **2** | **2** | **2** | **-** | **-** | **-** | **-** | **1** | **-** | **-** | **1** | **2** | **-** | **-** | **-** |
| [CC 2104.3] | Describe and Implement various inheritance and polymorphism forms using Java Classes and Interfaces. | **3** | **2** | **2** | **1** | **-** | **-** | **-** | **-** | **1** | **-** | **-** | **1** | **3** | **-** | **-** | **-** |
| [CC 2104.4] | Implement various collection data structure such as linked lists, queues, stacks using Java’s collection framework | **3** | **2** | **2** | **1** | **-** | **-** | **-** | **-** | **1** | **-** | **-** | **1** | **2** | **-** | **-** | **-** |
| [CC 2104.5] | Apply, Learn and finally implement the use of advanced programming constructs/ features such as exception handling, multithreading and event handling in real-life programming domains. | **3** | **2** | **2** | **2** | **-** | **-** | **-** | **-** | **1** | **-** | **-** | **1** | **2** | **-** | **2** | **3** |
| [CC 2104.6] | Visualize a real world problem in the form of various collaborating classes and objects for enhancing employability. | **1** | **2** | **1** | **1** | **-** | **-** | **-** | **-** | **1** | **-** | **-** | **2** | **2** | **-** | **3** | **3** |

1. **Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation**