upes-new-logo

**Format No. QSP/7.1/01.F01 (B) Issue No.05 Rev. No 5 Dated: Jan 1, 2017**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**College of Engineering Studies**

**Dehradun**

**COURSE PLAN**

Programme : B. Tech in (CSE+ All IBM Branches)

Course : OBJECT ORIENTED PROGRAMMING

Subject Code : CSEG 2016

No. of credits : 4

Semester : IV

Session : January 2020

Batch : 2018-2022

Prepared by : Pushpendra Kumar Rajput

Email : pushpendra@ddn.upes.ac.in

**Approved by HOD**

UPES Campus Tel : +91-135-2770137

“Energy Acres” Fax : +91 135- 27760904

P.O. Bidholi, , Dehradun

**COURSE PLAN**

1. **PREREQUISITE:**

* Knowledge of Basic Programming Language like C

1. **PROGRAM OUTCOMES (POs) for B.Tech in Computer Science and Engineering with Specialization in IFM :**

After completion of the program the student will be able to:

**PO1: Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems**: 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.

**PO5: Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society**: 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.

**PO7: Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: 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 multidisciplinary 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.

**PROGRAM SPECIFIC OUTCOMES**

**PSO1:** Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimizing techniques.

**PSO2:** Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and platforms.

**PSO3:** Apply the understanding of DevOps as cultural philosophies, practices, and tools that increase the ability to deliver applications and services at high velocity.

1. **OBJECTIVE OF COURSE:-**

The objective of this course is to:

1. Understand Java programs that leverage the object-oriented features.
2. Learn features of a strongly typed language: variable declaration and type compatibility checking.
3. Create Java classes by extending existing Java classes
4. Implement Interfaces in Java application
5. Apply Exception Handling mechanism in Java application.
6. Design, implement multithreading, and Data Structure.
7. Learn the concepts of JDBC, JSP, and Servlets.
8. **COURSE OUTCOMES (COs) for B.Tech in Computer Science and Engineering with Specialization in AI/ML:**

|  |  |
| --- | --- |
| CO1 | Understand complete architecture of Java Platform and able to relate with OOPS concepts |
| CO2 | Analyze real world object-oriented concepts and incorporate into the Java programming language. |
| CO3 | Implement and execute programs which are based on the OOPS concept. |
| CO4 | Differentiate Java Programming from other OOPS based programming languages |
| CO5 | Develop basic level applications based on core java and advance java skills. |

**Table: Correlation of POs v/s COs**

**Table: Correlation of POs and PSOs v/s COs**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PO/CO | PO  1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO  9 | PO  10 | PO  11 | PO  12 | PSO  1 | PSO  2 | PSO  3 |
| CO1 | 3 | 1 | 1 |  | 1 | 1 |  |  | 1 |  |  | 2 | 1 | 2 | 2 |
| CO2 | 2 | 3 | 2 | 2 | 1 | 1 | 1 |  | 1 | 1 |  | 2 | 2 | 2 | 2 |
| CO3 | 2 | 1 | 2 | 1 | 2 | 1 |  |  | 1 |  | 2 | 2 | 2 | 3 | 3 |
| CO4 | 2 | 2 | 1 |  | 1 | 1 |  |  | 1 |  |  | 2 | 1 | 2 | 2 |
| CO5 | 3 | 2 | 3 | 2 | 2 | 2 |  | 1 | 2 | 1 | 2 | 2 | 2 | 3 | 3 |

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Engineering Knowledge | Problem analysis | Design/development of solutions | Conduct investigations of complex problems | Modern tool usage | The engineer and society | Environment and sustainability | Ethics | Individual or team work | Communication | Project management and finance | Life-long Learning | Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimizing techniques | Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and platforms. |  |
| Course Code | Course Title | PO1 | PO2 | PO3 | PO 4 | PO 5 | PO6 | PO 7 | PO8 | PO9 | PO 10 | PO 11 | PO12 | PSO1 | PSO2 | PSO3 |
| CSEG  2016 | OOPs using Java | 1.8 | 2.4 | 2.0 | 0.0 | 1.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 0.0 | 0.0 |

|  |  |  |
| --- | --- | --- |
| **Mapping between COs and POs** | | |
| **CO1** | Memorize basic concepts of object oriented programming. | **PO1 PO2 PO3 PO5 PSO1** |
| **CO2** | Demonstrate real world object-oriented concepts and incorporate into the Java programming language. | **PO1 PO2 PO3 PO5 PSO1** |
| **CO3** | Execute programs, which are based on the concept of packages, interfaces and Exceptions. | **PO1 PO2 PO3 PO5 PSO1** |
| **CO4** | Implement the concepts of Strings, Threads and collections. | **PO1 PO2 PO3 PO5 PSO1** |
| **CO5** | Develop Java programs based on JDBC, JSP and servlets. | **PO1 PO2 PO3 PO5 PSO1** |

1. **COURSE OUTLINE**

**Unit I: Introduction**

**Unit II: Inheritance and Interfaces**

**Unit III: Packages, Exception and Wrapper Class**

**Unit IV: String Handling and Nested Classes**

**Unit V: Threads**

**Unit VI: JDBC, JSP, and Servlets**

1. **PEDAGOGY**

* Quiz
* Assignments
* Tests

1. **COURSE COMPLETION PLAN**

|  |  |
| --- | --- |
| Total Sessions | 48 |
| Total Quiz | 02 |
| Total Assignments | 02 |
| Total Tests | 02 |

One Session =60 minutes

1. **EVALUATION & GRADING**

Students will be evaluated based on the following 3 stages.

Internal Assessment - 30%

Mid-Term Examination - 20%

End-Term Examination - 50%

**H1. INTERNAL ASSESSMENT: WEIGHTAGE – 30%**

Internal Assessment shall be done based on the following:

|  |  |  |
| --- | --- | --- |
| S. No. | Description | % of Weightage out of 30% |
| 1 | Tests (2) and Quizzes (2) | 50% |
| 2 | Assignments (2) | 40% |
| 3 | Attendance | 10% |

Each Assessment is carried out for suitable marks and finally reduced suitably based on its weightage. At course completion, the student is awarded with the grade based on the composite score obtained out of 100 marks (30% IA + 20% MS + 50% ES).

**H2. MID TERM EXAMINATION: WEIGHTAGE – 20%**

Mid Term examination shall be of Two hours duration and shall be a combination of objective and true-false type questions.

***Date of showing Mid Term Examination Answer Sheets: Within a week after completion of mid Sem examination.***

**H3. END TERM EXAMINATION: WEIGHTAGE – 50%**

End Term Examination shall be Three Hours duration and shall be a combination of Short and Long theory/numerical Questions.

**H4. GRADING:**

The overall marks obtained at the end of the semester comprising all the above three mentioned shall be converted to a grade.

1. **SESSION PEDAGOGY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module/Session** | **Topic** | **Course Outcomes Addressed** | **Required Learning Resource (including media)** | **Discussion(s)/Postings** | | **Assignments/Quizzes** | |
| **Unit-1**  **(4 Hours)** | **Introduction** | **CO 1** |  |  | |  | |
| 1 | Feature of Java, JVM, JRE, setting Class path (Java Environment)  Classes, Fields, Access Control, Creating Objects |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 2 | Construction, Initialization, Methods, this, Overloading Methods, main Method, Native Methods, Class Design. |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 3 | **Lexical Elements, Types and Literals, Variables, Array Variables, Naming** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 4 | **Operators, Expressions, Member Access, Precedence, Associativity**, **Statements & Blocks, if-else,  switch, while and do-while, for, Labels, break, continue, return, goto** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| **Unit-2**  **(7 Hours)** | **Inheritance, Interfaces and Packages,** | **CO 2** |  |  | |  | |
| 5 | Extended Class, Constructors in Extended classes, Inheriting and Redefining Members |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 6 | Type Compatibility and Conversion, protected, final Methods and Classes, Abstract methods and classes |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 7 | **Object Class, Designing extended classes, Single Inheritance versus Multiple Inheritance** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 8 | **Interface, Interface Declarations, Extending Interfaces** | **CO3** | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 9 | Working with Interfaces, Marker Interfaces, When to Use Interfaces |  | Resources on Blackboard | Lectures and Discussions | |  | |
| 10 | More Examples and Practices on Interfaces |  | **Text Books and Reference Books** | Lectures and Discussions | |  | |
| 11 | **Package naming, type imports, package access** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 12 | Package contents, package objects and specifications, examples |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 13 | More Examples and Practices on Packages |  | **Text Books and Reference Books** | Lectures and Discussions | |  | |
| **Unit-3**  **(9 Hours)** | **Exception and String Handling, Wrapper Class** | **CO3** |  |  | | **Assignment-1** | |
| 14 | Creating exception types, throw, throws |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 15 | Try, catch and finally |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 16 | **Custom exception, when to use exception** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 17 | More Examples and Practices on Exceptions |  | **Text Books and Reference Books** | Lectures and Discussions | |  | |
| 18 | Wrapper classes and loading classes |  | **Text Books and Reference Books** | Lectures and Discussions | |  | |
|  |  |  |  |  | |  | |
| 19 | String operations, string comparisons, utility methods |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 20 | Making related strings, string conversions, strings and char arrays, string and byte arrays |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 21 | **StringBuffer, StringBuilder** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 22 | More Examples and Practices on Strings |  | **Text Books and Reference Books** | Lectures and Discussions | |  | |
| 23 | Quiz 1 |  |  |  | | **Quiz1** | |
| **Unit-4**  **(8 Hours)** | **Nested class and Threads** | **CO 4** |  |  | |  | |
| 24 | Static Nested Types, Inner Classes |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 25 | Local Inner Classes, Anonymous Inner Classes |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 26 | Inheriting Nested Types, Nesting in Interfaces, Implementation of Nested Types |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 27 | Creating Threads , Using Runnable, Synchronization |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 28 | Wait, notifyAll, and notify, Waiting and Notification, Thread Scheduling |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 29 | Deadlocks , Ending Thread Execution, volatile |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 30 | **Thread Management, Security, and ThreadGroup, Threads and Exceptions, debugging threads** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 31 | More Examples and Practices on Threads |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| **Unit-5**  **(9 Hours)** | **Collections, Design Pattern and JDBC** | **CO 4** |  |  | | **Assignment-2** | |
| 32 | Collections, Iteration, collection interface |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 33 | Set and SortedSet, List |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 34 | Map and SortedMap, Wrapped Collections and Collections Class |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 35 | **Arrays, Legacy Collection, Properties** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 36 | Design Pattern: Object Composition Principles Singleton Design Pattern |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 37 | Design Pattern:  Factory Design Pattern |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 38 | Design Pattern:  DAO Design Pattern |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 39 | Structure of JDBC program, Types of driver, driver manager class |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 40 | **JDBC statement: prepared, callable, Types of result set, Inserting and updating records** |  | **Text Books and Reference Books** | | Lecture and Discussions | |  | |
| **Unit-6**  **(8 Hours)** | **Advanced Java** | **CO 5** |  |  | |  | |
| 41 | Simple Servlet, Servlet life cycle, servlet program |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 42 | **Generic servlet, HTTP servlet** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 43 | **Servlet config, servlet context, Get and Post methods** |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 44 | JSP: Basic JSP architecture |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 45 | Lifecycle of JSP, JSP elements |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 46 | JSP directives: page directive. include directive |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 47 | JSP declaration, expression, scriplets |  | **Text Books and Reference Books** | Lecture and Discussions | |  | |
| 48 | Quiz 2 |  |  |  | | **Quiz-2** | |

1. **Text Books**

**T1:** The Java Programming Language 3rd Edition, Ken Arnold, James Gosling, Pearson

**T2:** Head First Servlets and JSP 2nd Edition.

**T3:** The Complete Reference Java 7th Edition, Herbert-Schild, TMH.

**T4:** Java SE7 Programmer I &II Study Guide, Kathy Sierra and Bert Bates, McGraw Hill.

1. **Reference Books**

**R1:** A premier guide to SCJP 3rd Edition, Khalid Mughal, Pearson

**R2:** Thinking in Java, 3rd Edition, Bruce Ackel, Pearson

1. **Web References:**
2. youtube.com
3. nptel.iitc.ac.in

**GUIDELINES**

***Cell Phones and other Electronic Communication Devices*:** Cell phones and other electronic communication devices (such as Blackberries/Laptops) are not permitted in classes during Tests or the Mid/Final Examination. Such devices MUST be turned off in the class room.

***E-Mail and online learning tool:*** Each student in the class should have an e-mail id and a pass word to access the LMS system regularly. Regularly, important information – Date of conducting class tests, guest lectures, via online learning tool. The best way to arrange meetings with us or ask specific questions is by email and prior appointment. All the assignments preferably should be uploaded on online learning tool. Various research papers/reference material will be mailed/uploaded on online learning platform time to time.

***Attendance:*** Students are required to have **minimum attendance of 75%** in each subject. Students with less than said percentage shall **NOT** be allowed to appear in the end semester examination.

**Course outcome assessment:** To assess the fulfilment of course outcomes two different approaches have been decided. Degree of fulfillment of course outcomes will be assessed in different ways through direct assessment and indirect assessment. In Direct Assessment, it is measured through quizzes, tests, assignment, Mid-term and/or End-term examinations. It is suggested that each examination is designed in such a way that it can address one or two outcomes (depending upon the course completion). Indirect assessment is done through the student survey which needs to be designed by the faculty (sample format is given below) and it shall be conducted towards the end of course completion. The evaluation of the achievement of the Course Outcomes shall be done by analyzing the inputs received through Direct and Indirect Assessments and then corrective actions suggested for further improvement.

***Passing criterion:***

* If the batch size is upto 30, Grading shall be done on the basis of absolute grading system
* If the batch size is more than 30, the grading will be done based on Relative Grading System

**Both in Absolute and Relative Grading System passing criteria will be:**

* **For UG-Students:** Scoring less than 35 absolute marks in individual course either in end semester examination or as composite score shall be awarded as ‘F’
* **For PG-Students:** Scoring less than 40 absolute marks in individual course either in end semester examination or as composite score shall be awarded as ‘F’
* **For UG & PG:** Students scoring 85 marks and above as composite score (IA+MS+ES) shall be awarded as the highest grade as ‘O’ i.e., Outstanding (on 10 point Scale) and ‘A’ i.e., Outstanding (on 4 point Scale)

**Sample format for Indirect Assessment of Course outcomes**

|  |
| --- |
| NAME: |
| ENROLLMENT NO: |
| SAP ID: |
| COURSE: |
| PROGRAM: |

Please rate the following aspects of course outcomes of Advance Programming Using Java

Use the scale 1-4\*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| COs |  | 1 | 2 | 3 | 4 |
| CO1 | Memorize basic concepts of object oriented programming. |  |  |  |  |
| CO2 | Demonstrate real world object-oriented concepts and incorporate into the Java programming language. |  |  |  |  |
| CO3 | Execute programs, which are based on the concept of packages, interfaces and Exceptions. |  |  |  |  |
| CO4 | Implement the concepts of Strings, Threads and collections. |  |  |  |  |
| CO5 | Develop Java programs based on JDBC, JSP and servlets. |  |  |  |  |

3

Below Average

Good

1

**\***

Very Good

Average

4

2