# COURSE LESSON PLAN

# COURSE LESSON PLAN

**Semester: IV Academic Year:2020-2021**

|  |  |
| --- | --- |
| Course Name**: Object Oriented Concepts** | Course Code: **18CS45** |
| Total Teaching Hours: **50 Hours** | Duration of Exam: **3 Hours** |
| Lecture-Tutorial-Practical (LTP): | Total No. of contact hours per week:**10 Hours** |
| Exam Marks: **60 Marks** | IA Marks: **40** |
| Lesson Plan Author: **Prapulla G** | Date: **20/03/2021** |
| Checked By: | Date: |

**Preamble:**

Most of the software needs to develop an application which runs in cross platform. Java is the one of the pioneer software tools used for cross platform development software. Java is the. most dominant software to develop web applications and distributed applications. This course provides basic concepts about Object Oriented Programming, Java Packages, Exceptions, Database connectivity, Networking, AWT and Java Servlets. After successful completion of this course learners can able to develop software modules for real world problems.

**Prerequisites:**

Problem Solving using C

**Course Outcomes:**

On completing this course, the students should have acquired the knowledge

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Outcome Number** | **Course Outcome Statement** | **Module**  **Number** | **Revised Bloom’s Level** |
| **CO1** | Explain the object-oriented concepts and JAVA | **1,2** | **L1, L2** |
| **CO2** | Develop computer programs to solve real world problems in Java. | **3,4** | **L1, L2, L3** |
| **CO3** | Develop simple GUI interfaces for a computer program to interact with users, and to comprehend the event-based GUI handling principles using applets and swings. | **5** | **L1, L2, L3** |

**Scheme from VTU website**

|  |
| --- |
| **Question Paper Pattern:**  • The question paper will have ten questions.  • Each full Question consisting of 20 marks  • There will be 2 full questions (with a maximum of four sub questions) from each module.  • Each full question will have sub questions covering all the topics under a module.  • The students will have to answer 5 full questions, selecting one full question from each module.  **Mandatory Note:**  Every institute shall organize bridge course on C++, either on vacation or in the beginning of even semester for a minimum period of ten days (2hrs/day). Maintain a copy of the report for verification during LIC visit. |

**COURSE CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULES Syllabus** | **Module Outcomes** | **TH** | **BTL** |
| **MODULE – 1**  Introduction to Object Oriented Concepts: A Review of structures, Procedure–Oriented Programming system, Object Oriented Programming System, Comparison of Object-Oriented Language with C, Console I/O, variables and reference variables, Function Prototyping, Function Overloading.  Class and Objects: Introduction, member functions and data, objects and functions. | **08** |  | **L1, L2** |
| **MODULE – 2**  **Class and Objects (contd):**  Objects and arrays, Namespaces, Nested classes, Constructors, Destructors.  **Introduction to Java: Java’s magic:** the Byte code; Java Development Kit (JDK); the Java Buzzwords, Object-oriented programming; Simple Java programs. Data types, variables and  arrays, Operators, Control Statements. | **08** |  | **L1, L2** |
| **MODULE – 3**  **Classes, Inheritance,Exception Handling:**  **Classes:** Classes fundamentals; Declaring objects; Constructors, this keyword, garbage collection.  **Inheritance:** inheritance basics, using super, creating multi-level hierarchy, method overriding.  **Exception handling:** Exception handling in Java. | **08** |  | **L1, L2, L3** |
| **MODULE – 4**  **Packages and Interfaces:** Packages, Access Protection, Importing Packages. Interfaces.  **Multi-Threaded Programming:** Multi-Threaded Programming: What are threads? How to make the classes threadbare; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, producer consumer problems. | **08** |  | **L1, L2, L3** |
| **MODULE – 5**  **Event Handling:** Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes.  **Swings:** Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon; JTextField;The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; JTable. | **08** |  | **L1, L2, L3** |

**\*TH- Teaching Hours, BTL-Blooms Taxonomy Level**

**REFERENCE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Book Type** | **Code** | **Title & Author** | **Publication Info** | | |
| **Edition** | **Publisher** | **Year** |
| **Text Book** | **T1** | Object Oriented Programming with C++,  Sourav Sahay | 2nd | Oxford University Press | 2006 |
| **T2** | Java The Complete Reference,  Herbert Schildt | 7th | Tata McGraw Hill | 2007 |
| **Reference Books** | **R1** | Programming with Java",  Mahesh Bhave and Sunil Patekar | 1st | Pearson  Education | 2008 |
| **R2** | The Complete Reference C++,  Herbert Schildt | 4th | Tata McGraw Hill | 2003 |
| **R3** | C++ Primer,  Stanley B.Lippmann, Josee Lajore | 4th | Pearson Education | 2005 |
| **R4** | Programming with Java A primer,  E Balagurusamy |  | Tata McGraw Hill companies. |  |
| **Web Resource (mention web link)** | **R5** | <https://www.w3schools.com/java/> | | | |

# Evaluation Scheme for INTERNAL ASSESSMENT

|  |  |
| --- | --- |
| **Assessment** | **Weightage in Marks** |
| Internal Assessment Exam 1 | 30 |
| Internal Assessment Exam 2 | 30 |
| Internal Assessment Exam 3 | 30 |
| Assignments/Any other activity | 10 |
| **Total** | **40** |

## Course Unitization for IA Exams and Semester Examination

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **IA Test 1** | **IA TEST 2** | **IA TEST 3** | **VTU SEE** |
| No of questions | No of questions | No of questions | No of questions |
| **MODULE 1** | 5 |  |  | 6 |
| **MODULE 2** | 5 | 4 |  | 6 |
| **MODULE 3** |  | 4 |  | 6 |
| **MODULE 4** |  |  | 4 | 6 |
| **MODULE 5** |  |  | 4 | 5 |

### **Note:** Each Question carries 20 marks and may consist of sub-questions.

**Date: Head of Department**

**MODULE WISE PLAN**

**MODULE -1**

**Lesson Schedule:**

|  |  |  |
| --- | --- | --- |
| **Class Number** | **Portion covered** | **Book** |
|  | Introduction to Object Oriented Concepts:  A Review of structures, Procedure–Oriented Programming system | T1 |
|  | Object Oriented Programming System | T1 |
|  | Comparison of Object-Oriented Language with C | T1 |
|  | Console I/O, variables and reference variables | T1 |
|  | Function Prototyping, Function Overloading. | T1 |
|  | Class and Objects: Introduction | T1 |
|  | member functions and data, | T1 |
|  | objects and functions. | T1 |

**Questions as per Bloom’s Taxonomy**

|  |  |  |
| --- | --- | --- |
| **Sl .No.** | **Level 1 Questions** | **CO** |
| **1** | List out the difference between procedure-oriented program and object-oriented program. | **CO1** |
| **2** | List and explain the features of Object-Oriented Program. | **CO1** |
| **3** | Explain "**this"** pointer with an example program. | **CO1** |
| **Sl .No.** | **Level 2 Questions** | **CO** |
| **1** | Define **function overloading** (or function polymorphism) and Illustrate with a C++ program for finding areas of circle (PI\*r\*r), rectangle(l\*b) and square (x\*x) by getting r.l,b,x through keyboard and printing the areas on console using the method Area() applying the concept of function overloading. | **CO1** |
| **2** | Explain friend function. And illustrate with a C++ program to add two complex numbers using **friend function** | **CO1** |
| **Sl .No.** | **Level 3 Questions** | **CO** |
| **1** | What is an inline function? Develop a C++ inline function to fin maximum of two numbers | **CO1** |

**MODULE -2**

**Lesson Schedule:**

|  |  |  |
| --- | --- | --- |
| **Class Number** | **Portion covered** | **Book** |
|  | Class and Objects (contd): Objects and arrays | T2 |
|  | Namespaces, Nested classes | T2 |
|  | Constructors, Destructors | T2 |
|  | Introduction to Java: Java’s magic: the Byte code | T2 |
|  | Java Development Kit (JDK); the Java Buzzwords | T2 |
|  | Object-oriented programming | T2 |
|  | Simple Java programs | T2 |
|  | Data types, variables | T2 |
|  | arrays, Operators | T2 |
|  | Control Statements. | T2 |

**Questions as per Bloom’s Taxonomy**

|  |  |  |
| --- | --- | --- |
| **Sl .No.** | **Level 1 Questions** | **CO** |
| **1** | Define Bytecode. | **CO1** |
| **2** | Define Type casting. | **CO1** |
| **3** | What are **static members of a class?** | **CO1** |
| **Sl .No.** | **Level 2 Questions** | **CO** |
| **1** | List and Explain Java buzz words. | **CO1** |
| **2** | With an example explain the working of >> and >>> operators in C++. | **CO1** |
| **3** | Explain portability in Java. | **CO1** |
| **4** | Explain Types of Type casting with an example | **CO1** |
| **Sl .No.** | **Level 3 Questions** | **CO** |
| **1** | Illustrate a C++ program to count the number of objects created. | **CO1** |
| **2** | Develop a program to show working of switch case. | **CO1** |

**MODULE -3**

**Lesson Schedule:**

|  |  |  |
| --- | --- | --- |
| **Class Number** | **Portion covered** | **Book** |
|  | Classes, Inheritance, Exception Handling:  Classes | T2 |
|  | Classes fundamentals | T2 |
|  | Declaring objects | T2 |
|  | Constructors, this keyword | T2 |
|  | garbage collection. | T2 |
|  | Inheritance: inheritance basics, using super | T2 |
|  | creating multi-level hierarchy | T2 |
|  | method overriding | T2 |
|  | Exception handling | T2 |
|  | Exception handling in Java | T2 |

**Questions as per Bloom’s Taxonomy**

|  |  |  |
| --- | --- | --- |
| **Sl.No.** | **Level 1 Questions** | **CO** |
| **1** | Define Inheritance. Explain the types of Inheritance in Java. | **CO2** |
| **2** | Define Exception. Explain with an example. | **CO2** |
|  |  | **CO2** |
| **Sl.No.** | **Level 2 Questions** | **CO** |
| **1** | Explain Java garbage collector. | **CO2** |
| **2** | Write a note on final keyword. | **CO2** |
| **3** | Distinguish between Method overloading and Method overriding in Java. | **CO2** |
| **Sl .No.** | **Level 3 Questions** | **CO** |
| **1** | Develop a program to show the use super keyword. | **CO2** |
|  |  |  |

**MODULE -4**

**Lesson Schedule:**

|  |  |  |
| --- | --- | --- |
| **Class Number** | **Portion covered** | **Book** |
|  | Packages and Interfaces: Packages, Access Protection | T2 |
|  | Importing Packages, Interfaces | T2 |
|  | Multi-Threaded Programming: Multi-Threaded Programming | T2 |
|  | What are threads? | T2 |
|  | How to make the classes threadable | T2 |
|  | Extending threads | T2 |
|  | Implementing runnable; Synchronization | T2 |
|  | Changing state of the thread | T2 |
|  | Bounded buffer problems | T2 |
|  | Producer consumer problems | T2 |

**Questions as per Bloom’s Taxonomy**

|  |  |  |
| --- | --- | --- |
| **Sl .No.** | **Level 1 Questions** | **CO** |
| **1** | What is Thread? Explain the two ways of creating a Thread in Java | **CO2** |
| **2** | What are Access Specifiers? Explain them. | **CO2** |
| **3** | Explain about the Delegation Event Model | **CO2** |
| **4** | Explain about the i) FocusEvent. ii) MouseEvent. | **CO2** |
| **Sl .No.** | **Level 2 Questions** | **CO** |
| **1** | Explain packages in Java with an example. | **CO2** |
| **3** | Explain Interfaces in java | **CO2** |
| **4** | Distinguish between multi-tasking and multithreading. How is synchronization and inter-thread communication done in java? | **CO2** |
| **5** | Explain about the adapter classes with suitable program. | **CO2** |
| **Sl .No.** | **Level 3 Questions** | **CO** |
| **1** | Develop a program to demonstrate Thread priorities in Java. | **CO2** |
| **2** | Develop a program to show synchronization in Java | **CO2** |

**MODULE -5**

**Lesson Schedule:**

|  |  |  |
| --- | --- | --- |
| **Class Number** | **Portion covered** | **Book** |
|  | Event Handling: Two event handling mechanisms | T2 |
|  | The delegation event model | T2 |
|  | Event classes; Sources of events | T2 |
|  | Sources of events; Event listener interfaces | T2 |
|  | Using the delegation event model, Adapter classes; Inner classes | T2 |
|  | Swings: Swings: The origins of Swing | T2 |
|  | Two key Swing features; Components and Containers | T2 |
|  | The Swing Packages; A simple Swing Application | T2 |
|  | Create a Swing Applet; Jlabel and ImageIcon; JTextField;The Swing Buttons | T2 |
|  | JTabbedpane; JScrollPane; JList; JComboBox; JTable | T2 |

**Questions as per Bloom’s Taxonomy**

|  |  |  |
| --- | --- | --- |
| **Sl .No.** | **Level 1 Questions** | **CO** |
| **1** | Write short notes on Event Listener Interfaces and explain any two interfaces with syntax. | **CO3** |
|  |  |  |
|  |  |  |
| **Sl .No.** | **Level 2 Questions** | **CO** |
| **1** | Describe key features of Swings in Java | **CO3** |
| **2** | Compare AWT and Swings. | **CO3** |
| **3** | Write a note on JFrame and JButton | **CO3** |
| **4** | List applet initialization and termination method? Write a java applet that set the background color cyan and foreground color red and output a string message “A simple Applet”? | **CO3** |
| **Sl .No.** | **Level 3 Questions** | **CO** |
| **1** | Create swing applet that has two buttons named beta and gamma. When either of the buttons pressed, it should display “beta pressed” and “gamma pressed” respectively. | **CO3** |
|  |  |  |

**Teaching-Learning Methods**

1. Lecture (Chalk and talk)
2. Demonstration (using components/models etc.)
3. Seminars
4. Concept map
5. Role play
6. Audio-video
7. Brain storming
8. Quiz
9. Group discussion and debate
10. Field trips/Industrial visit

Note: A few examples are given for reference. Map the planned activity in the session plan.

**Additional Academic documents**

Date:

**Vision**

To develop competent professionals with strong fundamentals in Information Science and Engineering, interdisciplinary research and ethical values for the betterment of the society.

**Mission**

**M1-** To establish a transformational learning ambience with good infrastructure facilities to impart knowledge and the necessary skill set to produce competent professionals.

**M2-** To create a new generation of engineers who excel in their career with leadership/entrepreneur qualities.

**M3-** To promote sustained research and innovation with an emphasis on ethical values.

**Program educational objectives of the program(PEO’s)**

**PEO1:**  To expertise in problem analysis, solving, design, development and necessary information to meet technical and managerial challenges.

**PEO2:**  To pursue interdisciplinary research and higher studies with profound knowledge enriched with academics and information technology skills.

**PEO 3:** To excel in competitive environment towards leadership and life-long learning for a successful professional career.

**Program Specific Outcomes (PSO’s)**

**PSO1:** Able to find the solutions to problems using programming languages by applying the suitable data and file structures concept and obtain optimal performance through design and analysis of algorithms.

**PSO2:** Able to work on the recent technologies by managing and organizing the processed data for new growth of opportunities in the industries exploring web designing and simulating the real world problems.

**PSO3:** Able to Apply the principles of software engineering to develop computational models under realistic constraints and ability to provide solutions based on the expertise in networking, database management and entrepreneurship to the various needs.

**Program outcomes**

At the end of the B.E program, students are expected to have developed the following outcomes.

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **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.
4. **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.
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.
6. **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.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
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 multidisciplinary environments.
12. **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

**Mapping & Justification of PO’s & PSO’s to CO’s**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **CO - PO matrix for each course** | | | | | | | | | | | | | | |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | **"-"** | **"-"** | **"-"** | **"-"** | **"-"** | **"-"** | **"-"** | **"-"** | **2** | **"-"** | **"-"** | **2** | **2** | **"-"** | **"-"** |
| **CO 2** | **"-"** | **"-"** | **3** | **"-"** | **2** | **"-"** | **"-"** | **"-"** | **2** | **"-"** | **"-"** | **2** | **2** | **"-"** | **"-"** |
| **CO 3** | **"-"** | **"-"** | **3** | **"-"** | **2** | **"-"** | **"-"** | **"-"** | **2** | **"-"** | **"-"** | **2** | **2** | **"-"** | **"-"** |
| **AVG VALUE** | **0** | **0** | **3** | **0** | **2** | **0** | **0** | **0** | **2** | **0** | **0** | **2** | **2** | **0** | **0** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CO - PO Mapping Justification** | | |
| **CO's** | **PO's** | **3/2/1/NA** | **Justification** |
| **CO1** | **PO9** | **2** | Group assignments were given to the students to understand the Oobject Oriented Concepts in JAVA, that will help in building teamwork among them |
| **PO12** | **2** | knowledge on programming languages will prepare the students to engage in lifelong learning. |
| **CO2** | **PO3** | **3** | Students will develop Object Oriented Programming for the real time problems. |
| **PO5** | **2** | To develop programs in java to solve problems modern tools like eclipse is used. |
| **PO9** | **2** | Group assignments were given to the students to solve some problems, that will help in building teamwork among them |
| **PO12** | **2** | Students will be engaged to learn latest versions of java independently |
| **CO3** | **PO3** | **3** | Able to use GUI and exception handlers in Java, design and develop programs using applets. |
| **PO5** | **2** | To develop GUI modern tools like eclipse is used. |
| **PO9** | **2** | Group assignments were given to the students to solve some problems, that will help in building teamwork among them |
| **PO12** | **2** | Developing GUI interfaces to help the students to engage in life learning. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CO - PSO Mapping Justification** | | |
|  | **PSO's** | **3/2/1/NA** | **Justification** |
| **CO 1** | **PSO1** | **2** | By using Object Oriented Concepts students can find solution for the problems C++ and Java programming languages. |
| **CO 2** | **PSO1** | **2** | With the help of Java programming students can analyse and design real world problems. |
| **CO 3** | **PSO1** | **2** | Students can develop simple GUI interfaces for JAVA programs |