Subject Code | Lecture | Tutorial | Practical | Credit
--- | --- | --- | --- | ---
TGB | 3 | 0 | -- | 3

TGB Object Oriented Programming with Java

Preamble
This course provides a comprehensive introduction to the object-oriented approach through the widely-used Java programming language. Java is a multipurpose, easy to learn, powerful platform where virtually a complete application in any domain can be implemented. Java is the most widely used language for developing open source software systems. With Java, there comes huge library support which reduces the development time and enhances software reuse for faster development. This course concentrates on object oriented aspects of Java and from this course the student will gain a solid basis for further study of the object-oriented software development.

Programming Outcomes Addressed
a. Graduates will demonstrate an ability to apply knowledge of engineering, information technology, mathematics and science.
b. Graduates will demonstrate an ability to design and conduct experiments, as well as to analyze and interpret data.
c. Graduates will demonstrate an ability to design a system or component, or process to meet stated specifications.
d. Graduates will demonstrate an ability to identify, formulate and solve engineering problems.
e. Graduates will demonstrate an ability to use techniques, skills, and modern engineering tools to implement and organize engineering works under given constraints.

Competencies
At the end of the course the student will be able to
1. Understand the concepts of object oriented programming like abstraction, encapsulation and polymorphism.
2. Implement exception handling mechanisms, packages, and interfaces in Java.
3. Create Java applications that are portable and secure.
4. Design applications for the Internet.

Assessment Pattern

| Bloom’s | Test 1 | Test 2 | Test 3/End |
--- | --- | --- | --- |
<table>
<thead>
<tr>
<th>Category</th>
<th>semester Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Remember</td>
<td>20 10 10</td>
</tr>
<tr>
<td>2 Understand</td>
<td>30 20 10</td>
</tr>
<tr>
<td>3 Apply</td>
<td>50 50 60</td>
</tr>
<tr>
<td>4 Analyze</td>
<td>0 0 0</td>
</tr>
<tr>
<td>5 Evaluate</td>
<td>0 0 0</td>
</tr>
<tr>
<td>6 Create</td>
<td>0 20 20</td>
</tr>
</tbody>
</table>

**Learning Objectives**

**Remember**
1. Define polymorphism.
2. What is a constructor?
3. How data encapsulation is achieved in Java?
5. What are abstract classes?
6. Define interface.
7. List and explain the different access modifiers.
8. What is the role of the finally block in exception handling?
9. What are I/O streams?
10. What are AWT controls?

**Understand**
1. What is the difference between a constructor and a member method?
2. List the similarities and differences between an abstract class and an interface.
3. How super keyword is used with respect to inheritance?
4. What are the three forms in which the final keyword can be used?
5. Differentiate method overloading and method overriding.
6. How will you execute a program in a package that used a class in another package?
7. How exception handling is done in Java.
8. What is the difference between Integer and int?
9. What is the difference between exception and error?
10. Demonstrate how throw and throws differ in Exception handling.

**Apply**
1. Write a recursive function in Java to generate a Fibonacci series.
2. Write a Java program that prints the numbers in the range [1 – 100] such that the difference between successive numbers increase by 1. That is it should print 1, 2, 4, 7, 11, 16, etc.
3. Create a applet that accepts two input string using <param> tag and concatenate the strings and display it in status window.
4. Write a program to i) convert lower case string to upper case ii) to compare two strings.
5. Let 'Series' be an interface with int sum(int n) as its method. Let class1 and
class2 be the two classes implementing the 'Series' interface to perform sum of
numbers upto n and squared sum of numbers upto n respectively. From the main
class use both the implementations through an interface reference.

6. Write a JAVA program to implement the Employee Details in a company.
   Let employee be the parent class with name, designation, no of days worked.
   Write an abstract method to compute salary depending on the no of days worked.
   Derive two subclasses Clerk and Manager.
   Override the salary computation method in both the subclasses.
   Use constructor overloading in the parent class.
   (Note: For salary computation include HRA, DA, Medical Allowance for clerk and
   Manager separately)

7. Write a JAVA program to implement the activities in the BANK.
   Let bank be the parent class with account no, name.
   Write an abstract method for interest calculation.
   Derive three subclasses Fixed Deposit, Variable Recurring Deposit and Savings
   account transactions.
   Override the interest calculation method in all the subclasses.
   Let the Saving account class has withdraw and deposit methods in addition to the
   interest calculation method. Use method overloading also. (Note: Assume
different interest rates for the different types).

Create

1. Create the Client/Server application to calculate the factorial of a number using
   Remote Method Invocation in JAVA.

2. Design an interface Queue, with methods to add and remove elements (integers).
   Furthermore, a method to check whether the queue is empty or not should exist.
   Implement the queue with an array. If the array becomes too small to hold all
   added elements, create a new larger (double the size of instance) array and copy
   all elements of the small array to the new one.

3. For a Bank application create a user defined exception subclass for the following
   conditions (i) Account holder’s age should not be less than 18, if it is less then a
   joint account holder has to be specified (ii) Minimum balance after withdrawal
   should not be less than 500. From the main class, create the account and do
   withdrawal operation.

4. Design a web page using AWT components and handle the events generated by
   each of them.

5. Create two packages pack1 and pack2 for an airline reservation system. Let the
   Flight’s private details like flight name, no., fare be in the pack1 and reservation
details like no. of seats available in pack2. From pack2 get the no. of tickets required, check its availability and calculate the fare.

6. Create a calculator using the AWT controls to get two numbers, perform one of the operations like addition, subtraction, multiply and divide and display the result.

Concept Map

Course Contents and Lectures schedule

<table>
<thead>
<tr>
<th></th>
<th>Genesis and Overview of JAVA</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Programming Constructs</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Data types, Variable, Arrays</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Operators and Expressions</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Control Flow Statements</td>
<td>2</td>
</tr>
<tr>
<td>1.4</td>
<td>String Handling</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Object Oriented Constructs</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Encapsulation - Classes and Objects</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Polymorphism - Constructor Overloading, Method Overloading</td>
<td>3</td>
</tr>
<tr>
<td>2.3</td>
<td>Inheritance</td>
<td>3</td>
</tr>
<tr>
<td>2.4</td>
<td>Interfaces</td>
<td>2</td>
</tr>
<tr>
<td>2.5</td>
<td>Packages</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Exception Handling</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Exception subsystems</td>
<td>3</td>
</tr>
<tr>
<td>3.2</td>
<td>Built-in Exceptions</td>
<td>2</td>
</tr>
<tr>
<td>3.3</td>
<td>User Defined Exceptions</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>File Handling</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Applications</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Client Server Application</td>
<td>2</td>
</tr>
</tbody>
</table>
5.2 Embedded Application

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Graphical User Interface Design - Applets</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Hours</td>
</tr>
</tbody>
</table>

Syllabus


Text Book


References


Web Resources on Java

1. Tutorial on Java [http://www.java2s.com/Tutorial/Java/CatalogJava.htm](http://www.java2s.com/Tutorial/Java/CatalogJava.htm)

Course Designer

1. J. Jennifer Ranjani jenni@tce.edu