| Unit – 1 | | |
|---|---|---|
| Day – 1 | | |
| **Lec #** | **Topics to be covered :** | **Assignments :** |
| 1 & 2 | Object-Oriented Programming concepts (object, class, encapsulation, abstraction, inheritance, polymorphism, message passing, dynamic binding) | **Essential Assignment:**<br>1. Understanding Java Development Kit (JDK) and environment variables settings.<br>2. Write the "Hello World" - first Java Program. Compile and execute it.<br>3. Write comments (Author, date, objective of the program, main function) in the above program.<br>4. Create following arrays and assign incremental values starting from 1 to each element.<br>    1. Array with 2 rows and 3 columns.<br>    2. Array with 2 rows. First row has 3 cells, and second row has 2 cells.<br>    Print the content of both the arrays.<br>5. Make a small report of 3 pages on difference between C & Java.<br>6. Write a program to print your name and address.<br><br>**Desirable Assignment:**<br>1. Refer the String class from Javadoc. Write program which will apply any 10 methods of the String class.<br>2. Refer the StringBuffer class from Javadoc. Write program which will apply any 5 methods of the StringBuffer class. |

| Unit – 1 | | |
|---|---|---|
| Day – 2 | | |
| **Lec #** | **Topics to be covered :** | **Assignments :** |
| 3 | Features of Java, Java's magic: bytecode | **Essential Assignment:**<br>1. Write a program to perform addition of two numbers and display its answer. (No need to take user input)<br>2. Write a program to perform addition of two given numbers and display its answer. (With user input)<br>3. Write a program to perform addition of numbers between 3 to 10 using for loop.<br>4. Write a program to find the maximum from two numbers. (Without user input) |

| 4 | **Class:** Class fundamentals, the general form of a class, defining a class, creating objects, working with methods | 5. Check the working of the following code and give comments with respect to use of "outer" :<br><br>```java<br>outer:<br>    for (int i = 0; i < 10; i++) {<br>        for (int j = 0; j < 10; j++) {<br>            if (j == 1)<br>                break outer;<br>            System.out.println(" value of j = " + j);<br>        }<br>    } // end of outer loop<br>```<br>6. Write a program to create three objects of Student class(name, age). Write getDetails method which takes user inputs (name and age). Write other method named displayDetails to display the name and age of the user. Create three objects and test working of above two methods.<br><br>**Desirable Assignment:**<br>1. Modify above Student class program to find the student having highest age among 3 students.<br>2. Write a program to create three objects of Book class(book name, author , price). Write a method getDetails which takes user inputs (book name , author and price). Write other method name displayDetails to display the book name , author and price. Create three objects and test working of above two methods.<br>3. Write a program to perform addition of numbers between 3 to 10 using while loop and do while loop.<br>4. Write a program to perform addition of numbers between 3 to 10, except number 5.<br>5. Write a program for calculator using switch case for addition & subtractions operations. |

| Unit – 1 |
|---|
| Day – 3 |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|

| 1 | Constructors (default, parameterized, copy constructor) | **Essential Assignment:** <br> 1. Modify the Student class to add default, parameterized & copy constructors. Create three objects of the Student class to test the above three constructors. Print the content of the all the three objects. <br> 2. Modify the above program to jump from one constructor to other constructor. <br> 3. Add initializer and static block in the above program. <br><br> **Desirable Assignment:** |
|---|---|---|
| 2 | Garbage collector, this keyword, static block, initializer block | 4. Modify the Book class to add default, parameterized & copy constructors. Create three objects s to test the above three constructors. Print the content of the all the three objects. <br> 5. Modify the above program to jump from parameterized constructor to default constructor. Execute the program. <br> 6. Add initializer and static block in the above program. Check the execution sequence of initializer, static block and constructor. |

|  |
|---|
| **Unit – 1** |
| **Day – 4** |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 | Passing object to the method, returning object from the method, method overloading, constructor overloading | **Essential Assignment:** <br> 1. Modify the Student class to add a method which will compare two other Student's  age and print the name of student with highest age. <br> 2. Modify the above method so that it returns the Student's object having the highest age. Print the name of the student and test your program. Test your program. <br> 3. Create a Calculator class to add two int and to add two floats. Use method overloading. |
| 2 | Static variable, static method, nested and inner classes, local class, anonymous class | 4. In the Student class add static variable to store the number of the objects created. Display the value of the count using the static method. Test your program. <br> 5. Create an Outer class with instance variable named "a". Inside this Outer class, create an Inner class. This inner class must have instance variable named "b". Assign some initial values to this "a" and "b". Create object of Outer class and print values of "a" and "b". <br><br> **Desirable Assignment:** <br> 6. Modify the Student class to add a static method which will compare two Student's object's age and |

|  |  | return the object with highest age. |
|---|---|---|
|  |  | 7.  Write a calculator program to add, subtract, multiply and division of two integers, two floats and two double number using method overloading and static methods. |

| Unit – 2 |||
|---|---|---|
| Day – 5 |||
| **Lec #** | **Topics to be covered :** | **Assignments :** |
| 1 & 2 | Inheritance, constructors and inheritance, types of inheritance, uses of super keyword for constructor and method | **Essential Assignment:**<br>1.  Create following classes:<br>    a.  Emp(id,name,age)<br>    b.  PartTimeEmp(numberOfHours,ratePerHour) inherits Emp<br>    c.  FullTimeEmp(basicPay,DA) inherits Emp<br>In all the three classes, add default and parameterized constructors. Create one object of both the child classes and print the content of both the objects.<br><br>In the all the three classes, add getDetails and showDetails methods. Create one object of both the child classes and print content of both the objects.<br><br>**Desirable Assignment:**<br>2.  Create following classes:<br>    a.  Account(id,name,balance)<br>    b.  SavingAccount(rateOfInterest) inherits Account<br>    c.  CurrentAccount(overDraftLimit) inherits Account<br>In all the three classes, add default and parameterized constructors. Create one object of both the child classes and print the content of both the objects. |

| | | In the all the three classes, add getDetails and showDetails methods. Create one object of both the child classes and print content of both the objects. |
|---|---|---|
| | | |

<table>
<tr><td colspan="3" align="center"><b>Unit – 2</b></td></tr>
<tr><td colspan="3" align="center"><b>Day – 6</b></td></tr>
<tr><td><b>Lec #</b></td><td><b>Topics to be covered :</b></td><td><b>Assignments :</b></td></tr>
<tr>
<td>1 & 2</td>
<td>Polymorphism, abstract class, final keyword</td>
<td>

**Essential Assignment:**
1. In the day-5 program, create object / reference of Emp class which will point to any of the child class. Call the getDetails and showDetails methods and check working of polymorphism.
2. In the day-5 program, make the Emp class as an abstract class. Add calculateSalary as abstract function. Try to create object of this class.

**Desirable Assignment:**
1. In the day-5 program, create object /reference of Account class which will point to any of the child class. Call the getDetails and showDetails methods and check working of polymorphism.
2. In the day-5 program, make the Account class as abstract class by adding appropriate abstract function. Try to create object of this class.

</td>
</tr>
</table>

<table>
<tr><td colspan="3" align="center"><b>Unit – 2 & 3</b></td></tr>
<tr><td colspan="3" align="center"><b>Day – 7</b></td></tr>
<tr><td><b>Lec #</b></td><td><b>Topics to be covered :</b></td><td><b>Assignments :</b></td></tr>
</table>

| 1 & 2 | Object class and its methods, interface, working with interface, multiple inheritance with interface | **Essential Assignment:** |
|---|---|---|
| | | 1. For the Student class(id,name,age) override toString and  equals methods of object class. Test working of both the methods. |
| | | 2. Create Interface Ione with class constant a having initial value 10. |
| | | Create Interface Itwo with class constant b having initial value 20. |
| | | Create Interface Ithree with class constant c having initial value 30. |
| | | Interface Ithree must inherit Ione & Itwo (multiple inheritance). |
| | | Crete class Test which implements Ithree. Test your program by creating an object of Test class and printing contents of a, b and c class constants. |
| | | **Desirable Assignment:** |
| | | 3. In day-5 program (Desirable Assignment section) add an interface IEmp with getDetails and showDetails as two abstract functions. Implement this interface on Emp class. Other coding will be same. Check working of the interface concept. |

| Unit – 3 |
|---|
| Day – 8 |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Exception hierarchy, exception handling fundamentals, the consequences of an uncaught exception, using multiple catch statements, catching subclass exception, nested try blocks, re-throwing an exception, finally & throws keywords, java's built-in exceptions, user define exception | **Essential Assignment:** |
| | | 1. Write a program which will handle the  ArrayIndexOutOfBoundsException and DivideByZeroException with multiple catch and finally block. |
| | | 2.  For the Student class (id,name,age), write a program to raise an user define exception for the age less than 20 years. |
| | | **Desirable Assignment:** |
| | | 3. Modify the above program of Student Class by taking the age from user using method. |
| | | 4. Test nested exception by putting exception handling mechanism in the above method and also in main method. |

| Unit – 3 |
|---|

| Day – 9 | |
|---|---|
| **Lec #** | **Topics to be covered :** | **Assignments :** |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Collection, Collection framework, Collection interface, Iterator interface, list interface, stack interface,  vector interface<br><br>**Difficulty Solving for Day1 to Day 9.** | **Essential Assignment:**<br>1.  With the help of Student(id,name,age) class do the followings:<br>    a.  Create list of 5 students. Iterate from first to last student and display all the information about each student.<br>    b.  Create stack of 5 students. Iterate from first to last student and display all the information about each student.<br>    c.  Create vector of 5 students. Iterate from first to last student and display all the information about each student.<br><br>**Desirable Assignment:**<br>2.  With the help of Book(id,name,price) class do the followings (Refer Javadoc):<br>    d.  Create Queue of 5 Books. Iterate from first to last book and display all the information about each book.<br>    e.  Create SortedSet of 5 Books Iterate from first to last book and display all the information about each book.<br>    f.  Create TreeSet of 5 Books. Iterate from first to last book and display all the information about each book.<br>**Difficulty Solving for Day1 to Day 9.** |

| Unit – 4 | |
|---|---|
| **Day – 10** | |
| **Lec #** | **Topics to be covered :** | **Assignments :** |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | **Difficulty Solving for Day1 to Day 9.** | 1.  **Difficulty Solving for Day1 to Day 9.** |

| Unit – 4 |
|---|
| **Day – 11** |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Multithreading fundamentals, the thread class and runnable interface, creating a thread, creating multiple threads, determining thread methods, thread priorities | **Essential Assignment:**<br>2. Write a multi-threaded program in which two threads prints 1 to 5000 numbers by extending Thread class.<br>3. Write a multi-threaded program in which two threads prints 1 to 5000 numbers by implementing Runnable interface.<br><br>**Desirable Assignment:**<br>1. Write a two threaded program in which first thread prints 1 to 5000 numbers and second thread prints 5000 to 1 numbers. |

| Unit – 4 |
|---|
| Day – 12 |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Need for synchronization, thread synchronization. Working with File Class. Creating File & Folder, Renaming File & Folder, Deleting Folder | **Essential Assignment:**<br>1. Write a multi-threaded program in which two threads access the common variable count which is initialize with 0. Both the read this count and print on console along with thread name. After printing, it will increment the content of this count variable. Both thread will do this work for 5000 times. Use the synchronization for count increment logic.<br>2. Write a program to create a new folder, rename and delete it.<br><br>**Desirable Assignment:**<br>3. Write a program to list all the files or directories available in given directory. |

| Unit – 4 |
|---|
| Day – 13 |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Byte streams and character streams, the byte stream classes, the character stream classes, the predefined streams, using the byte streams (reading console input, writing console output) | **Essential Assignment:**<br>1. Write a program to copy one file to other file using:<br>   1) Character by character<br>   2) Word by word<br>   3) Line by line<br>2. Write the program to marge two files in to third file using any of the above approaches.<br><br>**Desirable Assignment:**<br>3. Write a program to reverse the content of the file. File name must be given by user.<br>4. Write a program which will take one word from user which is supposed to be find in the file. Find the word in that file and display the list of indexed (position) of that word in that file. |

|  |
|---|
| **Unit – 4** |
| **Day – 14** |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Reading and writing files using byte streams (inputting from a file, writing to a file), automatically closing a file, reading and writing binary data, random access file | **Essential Assignment:**<br>1. Create Student class(Id,name,age). Put appropriate constructor to initialize all the data member of the class. Write 5 objects to the file. Id of the student must be 1,2,3,4 and 5. User will provide the Id of the student, and your program will display the information of that student.<br><br>**Desirable Assignment:**<br>2. Modify the above program-1 and provide the searching facility by using name and age. If multiple students found, display all the student's information one by one. |

|  |
|---|
| **Unit – 5** |
| **Day – 15** |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|

| 1 & 2 | GUI Programming, AWT Components, swing components, comparing AWT and swing, event handling mechanism<br><br>**Swing components with event handling**: JFrame, JPanel, Jbutton | **Essential Assignment:**<br>1. Create a JFrame with a button named "Click Me". Register JFrame with  MouseListener, MouseMotion Listener, KeyListener, WindowsListener. Print appropriate message which will show the execution of particular event of above listener. While clicking on button,  with help of ActionListener,<br><br>**Desirable Assignment:**<br>2. Create Adapter class for all the above listener and test working of it. |
|---|---|---|

| Unit – 5 |
|---|
| Day – 16 |

| Lec # | Topics to be covered : | Assignments : |
|---|---|---|
| 1 & 2 | Layout Managers (FlowLayout, BorderLayout, GridLayout, GridBagLayout, CardLayout), JLabel, JTextField, JTextArea | **Essential Assignment:**<br>1. Demonstrate the use of FlowLayout, BorderLayout, GridLayout, GridBagLayout & CardLayout.<br>2. Demonstrate working of JTextField, JTextArea with event handling.<br><br>**Desirable Assignment:**<br>3. Write a program for GUI based calculator with addition, subtraction, multiplication and division functionalities.<br>4. Write GUI based Notepad with following Menus:<br>File : New, Close,Exit<br>Edit : Cut,Copy,Paste<br>Help : About Notepad<br><br>Notepad must show the appropriate message on status bar after completion of any activity from above menu. |

| Unit – 5 |
|---|

| Day – 17 | | |
|---|---|---|
| **Lec #** | **Topics to be covered :** | **Assignments :** |
| 1 & 2 | JPasswordField, JCheckBox, JRadioButton, JComboBox, JList, Adapter classes, dialog boxes, working with menus | **Essential Assignment:**<br>1. Create GUI based application for storing student's information like (id,name,age,email,sem). Do the validation for valid email address. Form must have Submit and Clear button. On clicking on Submit button, all the information filled must be displayed by appropriate dialog box.<br>2. Convert the above program with following menus:<br>    a.  Operations : New, Close, Exit<br>    b.  Help: About Application<br><br>**Desirable Assignment:**<br>3. Modify the above program-3 to store the student's data in the file. Add one more button named "Find By ID", which helps the user to see the student's information by given ID.<br>4. Modify the above program-4 to provide the Next, Previous, First and Last button to display the students information. |

| Unit – 1,2,3,4,5 | | |
|---|---|---|
| **Day – 18** | | |
| **Lec #** | **Topics to be covered :** | **Assignments :** |
| 1 & 2 | Revision Session | **Completing pending work of Day1 to Day 17** |