



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

ELECTRONICS AND COMMUNICATION ENGINEERING

TUTORIAL QUESTION BANK

Course Title	JAVA PROGRAMMING				
Course Code	ACS552				
Programme	B.Tech				
Semester	VI	ECE			
Course Type	Core				
Regulation	IARE - R16				
Course Structure	Theory			Practical	
	Lectures	Tutorials	Credits	Laboratory	Credits
	3	-	3	-	-
Chief Coordinator	Mr. G Chandra Sekhar, Assistant Professor				
Course Faculty	Mr. G Chandra Sekhar, Assistant Professor				

COURSE OBJECTIVES:

The course should enable the students to:

I	Understand fundamentals of object-oriented terminology and programming concepts in java.
II	Acquire basics of how to translate solution problem into object oriented form.
III	Develop programs in java for solving simple applications.
IV	Design and implement simple program that use exceptions and multithreads.

COURSE OUTCOMES

CO 1	Able to learn the concept of object oriented programming that helps to organize complex programs
CO 2	Understand the appropriate roles of subtyping and inheritance, and use them effectively.
CO 3	Demonstrate an ability to design high speed, fault tolerant applications using multi-threading and exception handling concepts.
CO 4	Design and develop the java applications by using concepts of interfaces and packages.
CO 5	Experiment with the usage of files and database connectivity, to familiarize the advanced java programming skills and develop java based web applications.

COURSE LEARNING OUTCOMES:

Students, who complete the course, will have demonstrated the asking to do the following:

ACS552.01	Use object oriented programming concepts to solve real world problems.
ACS552.02	Explain the concept of class and objects with access control to represent real world entities.
ACS552.03	Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors.
ACS552.04	Describe the concept of operators and variables, arrays, parameter passing.
ACS552.05	Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords.
ACS552.06	Use dynamic and static polymorphism to process objects depending on their class.
ACS552.07	Analyze and understand the concept of abstract classes to define generic classes.
ACS552.08	Understand the impact of exception handling to avoid abnormal termination of program using checked and unchecked exceptions.
ACS552.09	Demonstrate the user defined exceptions by exception handling keywords (try, catch, throw, throws and finally).
ACS552.10	Use multithreading concepts to develop inter process communication.
ACS552.11	Understand the use of interrupting threads in the real world.
ACS552.12	Understand the importance of interfaces to develop real world java applications.
ACS552.13	Illustrate different techniques on creating and accessing packages (fully qualified name and import statements).
ACS552.14	Demonstrate the import statement usage and built-in packages.
ACS552.15	Understand and implement the concepts on file streams and operations in java programming for a given application programs.
ACS552.16	Understand text, byte, and character input/output streams.
ACS552.17	Describe the backend connectivity process in java program by using JDBC drivers.
ACS552.18	Develop java application to interact with database by using relevant software component (JDBC Driver).

TUTORIAL QUESTION BANK

UNIT – I				
OOPS CONCEPTS AND JAVA PROGRAMMING				
PART – A (Short Answer Questions)				
S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1	State the importance of Object Oriented Programming.	Remember	CO1	CACS552.01
2	Distinguish between procedural language and OOP's.	Understand	CO1	CACS552.01
3	Define class with an example.	Remember	CO1	CACS552.01
4	What is Inheritance? List out different types of inheritance.	Understand	CO1	CACS552.01
5	Define Polymorphism with necessary example.	Remember	CO1	CACS552.01
6	Distinguish between C, C++ and java.	Remember	CO1	CACS552.01
7	List out the features or buzzwords of java..	Remember	CO1	CACS552.01
8	Describe history of java.	Understand	CO1	CACS552.01
9	List out different data types used in java.	Remember	CO1	CACS552.03
10	Describe about object with an example.	Remember	CO1	CACS552.01
11	State and describe the scope and life time of variables.	Understand	CO1	CACS552.04
12	List and describe different types of operators.	Remember	CO1	CACS552.01
13	Classify different types of an array with an example.	Understand	CO1	CACS552.02

14	Describe the use of expressions with an example.	Remember	CO1	CACS552.01
15	Classify different enumerated types with an example.	Remember	CO1	CACS552.01
16	List out different types of control flow statements.	Understand	CO1	CACS552.03
17	List out the advantages of inheritance.	Remember	CO1	CACS552.04
18	Distinguish between constructor and method with an example.	Understand	CO1	CACS552.04
19	Demonstrate about data abstraction with an example.	Remember	CO1	CACS552.01
20	Describe about type conversion and type casting with a necessary examples	Understand	CO1	CACS552.01
Part - B (Long Answer Questions)				
1	Describe in detail about OOP's concepts with a necessary examples.	Understand	CO1	CACS552.01
2	Explain briefly about the features (buzzwords) of Java.	Understand	CO1	CACS552.01
3	Justify the statement "Java is a pure object oriented programming language"	Remember	CO1	CACS552.01
4	Describe briefly about the JVM architecture.	Understand	CO1	CACS552.01
5	Demonstrate briefly about the importance of "this" keyword with a necessary example.	Understand	CO1	CACS552.04
6	Describe the concept of method overloading with an example.	Remember	CO1	CACS552.04
7	State and describe constructor overloading with an example.	Understand	CO1	CACS552.04
8	State and describe different types of arrays with an example.	Understand	CO1	CACS552.03
9	Describe the usage of static block in the program with an example program.	Understand	CO1	CACS552.02
10	Discuss about various conditional statements in java with suitable examples.	Understand	CO1	CACS552.03
11	List out various types of variables in detail with an example.	Remember	CO1	CACS552.03
12	List out various ways for creating an object for a class.	Remember	CO1	CACS552.03
13	Describe the syntax of method in detail with an example.	Understand	CO1	CACS552.03
14	Compare and contrast narrowing and widening conversion in detail with an example.	Understand	CO1	CACS552.03
15	Describe about static method and variable with an example.	Remember	CO1	CACS552.04
Part - C (Problem Solving and Critical Thinking Questions)				
1	Predict the output of the code? Student john12 = new Student(1001, "John", 12); Student john13 = new Student(1002, "John", 13); System.out.println("comparing John, 12 and John, 13 with compareTo : " + john12.compareTo(john13));	Apply	CO1	CACS552.03
2	Interpret the output of the program. class Lifetime { public static void main(String args[]) { int x; for (x=0; x<3; x++) { int y=-1; System.out.println(" y is : " + y); y=100; System.out.println(" y is now : " + y); } } }	Evaluate	CO1	CACS552.03

3	<p>Predict output of the program.</p> <pre> public class If2 { static boolean b1, b2; public static void main(String [] args) { int x = 0; if (!b1) { if (!b2) { b1 = true; x++; if (5 > 6) x++; if (!b1) x = x + 10; } else if (b2 = true) x = x + 100; else if (b1 b2) x = x + 1000; } } System.out.println(x); } } </pre>	Apply	CO1	CACSS552.03
4	<p>Analyze the following code and give the output li valid or invalid.</p> <pre> public String getDescription(Object obj) { return obj.toString; } public String getDescription(String obj) { return obj; } public void getDescription(String obj) { return obj; } </pre>	Understand	CO1	CACSS552.03
5	<p>Predict the output of following program?</p> <pre> public class Test { public int aMethod() { static int i = 0; i++; return i; } public static void main(String args[]) { Test test = new Test(); test.aMethod(); int j = test.aMethod(); System.out.println(j); } } </pre>	Apply	CO1	CACSS552.02

6	<p>Identify the output of the program?</p> <pre> public class Test { public static void main(String args[]) { int i =1,j = 0; switch(i) { case 2: j += 6; case 4: j += 1; default: j += 2; case 0: j += 4; } System.out.println("j = " + j); } } </pre>	Remember	CO1	CACS552.03
7	<p>Predict the following program output.</p> <pre> Class Test { public static void main(String args[]) { int x, y; y=20; for(x=0; x<10: x++) { System.out.println("this is x:"+x); System.out.println("this is y:" +y); y= y-2; } } } </pre>	Apply	CO1	CACS552.04
8	<p>Identify output of the program?</p> <pre> Class BitShift { public static void main(String [] args) { int x = 0x80000000; System.out.print(x + " and "); x = x >>> 31; System.out.println(x); } } </pre>	Remember	CO1	CACS552.03
9	<p>Predict the program and find out the output.</p> <pre> Class Equals { public static void main(String [] args) { int x = 100; double y = 100.1; boolean b = (x = y); System.out.println(b); } } </pre>	Apply	CO1	CACS552.04

10	<pre> public class Sum { // Overloaded sum(). // This sum takes two int parameters public int sum(int x, int y) { return (x + y); } // Overloaded sum(). // This sum takes three int parameters public int sum(int x, int y, int z) { return (x + y + z); } // Overloaded sum(). // This sum takes two double parameters public double sum(double x, double y) { return (x + y); } // Driver code public static void main(String args[]) { Sum s = new Sum(); System.out.println(s.sum(10, 20)); System.out.println(s.sum(10, 20, 30)); System.out.println(s.sum(10.5, 20.5)); } } </pre>	Apply	CO1	CACS552.04
----	---	-------	-----	------------

UNIT – II

INHERITANCE

Part – A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1	Describe the importance of inheritance in object oriented programming.	Remember	CO2	CACS552.05
2	List out various types of inheritances in java.	Remember	CO2	CACS552.05
3	Define the term static binding.	Remember	CO2	CACS552.05
4	With an example, explain how to achieve run time polymorphism in java programming.	Remember	CO2	CACS552.06
5	Write about abstract class with its syntax.	Understand	CO2	CACS552.06
6	Interpret various member access rules in java.	Understand	CO2	CACS552.06
7	What is overriding in java explain with suitable examples.	Remember	CO2	CACS552.06
8	Explain in detail method overloading with suitable examples.	Remember	CO2	CACS552.06
9	How the final keyword is used to prevent inheritance in java.	Remember	CO2	CACS552.06
10	Discuss and define the syntax of Inheritance.	Understand	CO2	CACS552.06
11	Is it possible to achieve compile time polymorphism in java if yes explain with suitable example?	Understand	CO2	CACS552.07
12	Define and use final keyword to prevent overriding using example.	Remember	CO2	CACS552.07

13	Discuss with neat diagram explain the forms of inheritance.	Understand	CO2	CACS552.07
14	Describe multiple inheritance is not supported by Java.	Understand	CO2	CACS552.07
15	Differentiate between Inheritance and Encapsulation.	Understand	CO2	CACS552.06
16	Define dynamic binding with an example.	Remember	CO2	CACS552.06
Part - B (Long Answer Questions)				
1	Differentiate “this” and “super” keywords usage in java.	Understand	CO2	CACS552.05
2	List different types of inheritances supported by java with suitable examples.	Remember	CO2	CACS552.05
3	What is Object class. Discuss various methods of Object class.	Understand	CO2	CACS552.05
4	Illustrate the Use of “Super” keyword in method overriding with example.	Understand	CO2	CACS552.06
5	Compare and contrast interfaces and Abstract classes.	Understand	CO2	CACS552.06
6	Demonstrate dynamic binding with an example.	Understand	CO2	CACS552.06
7	List out the some of the standard overloaded methods in java.	Remember	CO2	CACS552.06
8	Describe Abstraction in java using abstract class with an example.	Remember	CO2	CACS552.06
9	Illustrate what happens if the parent and the child class have a field with same identifier.	Understand	CO2	CACS552.07
10	Why multiple inheritance not supported by java with suitable example.	Remember	CO2	CACS552.07
12	Compare and contrast overloading and overriding methods.	Understand	CO2	CACS552.07
14	State which method hides a method in the super class.	Remember	CO2	CACS552.07
15	Discuss the importance of final keyword in java with a program.	Understand	CO2	CACS552.06
16	State the benefits of inheritance with an example.	Remember	CO2	CACS552.07
17	Is it possible to override a private method in Java?	Understand	CO2	CACS552.06
18	List the rules of method over-riding in Java?	Remember	CO2	CACS552.07
Part – C (Problem Solving and Critical Thinking)				
1	<p>Interpret the program and give output.</p> <pre> class Animal { void eat() { System.out.println("eating..."); } } class Dog extends Animal { void bark() { System.out.println("barking..."); } } class TestInheritance { public static void main(String args[]) { Dog d=new Dog(); d.bark(); d.eat(); } </pre>	Apply	CO2	CACS552.06

5	<p>Predict the output of the program</p> <pre> class Demo { protected void getData() { System.out.println("Inside Demo"); } } class Demo1 extends Decmo { protected void getData() { System.out.println("Inside Demo1"); } } public class Test { public static void main(String[] args) { Demo obj = new Demo1(); obj.getData(); } } </pre>	Apply	CO2	CACS552.06
6	<p>Identify the output of the program?</p> <pre> class Super { public int i = 0; public Super(String text) { i = 1; } } class Sub extends Super { public Sub(String text) { i = 2; } } public static void main(String args[]) { Sub sub = new Sub("Hello"); System.out.println(sub.i); } } </pre>	Understand	CO2	CACS552.06
7	<p>Interpret and find out the output of the program?</p> <pre> public class Test { public int aMethod() { static int i = 0; i++; return i; } } public static void main(String args[]) { Test test = new Test(); test.aMethod(); int j = test.aMethod(); System.out.println(j); } } </pre>	Evaluate	CO2	CACS552.07

8	Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea().provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.	Remember	CO2	CACS552.07
---	--	----------	-----	------------

UNIT-III

EXCEPTION HANDLING AND MULTITHREADING

Part - A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1	Define the term Exception.	Remember	CO3	CACS552.08
2	Distinguish between exception and error.	Understand	CO3	CACS552.08
3	List out the benefits of exception handling.	Remember	CO3	CACS552.09
4	State the classification of exceptions.	Remember	CO3	CACS552.08
5	List out the checked exceptions.	Remember	CO3	CACS552.08
6	State the use of try and catch blocks with their syntax.	Remember	CO3	CACS552.08
7	Write about built in exception.	Understand	CO3	CACS552.09
8	What Is the Purpose of the Throw and Throws Keywords?	Remember	CO3	CACS552.09
9	Describe various unchecked exceptions.	Understand	CO3	CACS552.09
10	Define nested try and catch block with an example.	Remember	CO3	CACS552.08

11	Define thread in java. List out the various ways of creating thread.	Remember	CO3	CACS552.10
12	Describe the various states of threads.	Understand	CO3	CACS552.10
13	List the different ways to create a thread.	Remember	CO3	CACS552.10
14	Differentiate between throw and finally keyword with example.	Understand	CO3	CACS552.10
15	What is inter-thread communication?	Remember	CO3	CACS552.10
16	Illustrate the use of alive() and join() method with necessary examples	Remember	CO3	CACS552.10
17	List out the different thread priorities with example program.	Understand	CO3	CACS552.10
18	Distinguish between throw and throws.	Understand	CO3	CACS552.10
19	Define wait() state of the thread	Remember	CO3	CACS552.11
20	Justify the statement a “thread class can implements Runnable interface” with suitable example.	Remember	CO3	CACS552.11
21	Compare and contrast the differences between process and thread.	Remember	O3	CACS552.11

Part – B (Long Answer Questions)

1	Explain briefly about exception handling mechanism with suitable examples.	Understand	CO3	CACS552.08
2	Describe try, catch, and finally keywords used in exception handling with an example program	Remember	CO3	CACS552.08
3	Illustrate use of throws keyword with an example program	Understand	CO3	CACS552.08
4	Define a exception called “NotEqualException” that is thrown when a float value is not equal to 3.14. Write a program that uses the above user	Understand	CO3	CACS552.08
5	Differentiate between checked and unchecked exceptions.	Understand	CO3	CACS552.09
6	Exemplify the different types of exception.	Understand	CO3	CACS552.08

7	Illustrate builtin exceptions with suitable example.	Understand	CO3	CACS552.09
8	Explain how the user can throw user defined exception with example	Understand	CO3	CACS552.09
9	Describe the producer consumer problem with an example	Remember	CO3	CACS552.11
10	Explain with an example how java performs thread synchronization.	Understand	CO3	CACS552.10
11	Differentiate between multiprocessing and multithreading with a program.	Understand	CO3	CACS552.10
12	Explain briefly about the life cycle of a thread with a neat diagram.	Understand	CO3	CACS552.11
13	Interpret various methods of thread class.	Understand	CO3	CACS552.11
14	Describe a java program using thread priorities.	Remember	CO3	CACS552.10
15	Explain Daemon threads with an example.	Understand	CO3	CACS552.10
16	Exemplify the behavior of thread using thread class methods.	Understand	CO3	CACS552.11
17	Illustrate the process of creating thread by implementing Runnable interface.	Remember	CO3	CACS552.11
Part – C (Problem Solving and Critical Thinking Questions)				
1	<p>Identify the output of program</p> <pre> public class TestMultipleCatchBlock { public static void main(String args[]) { try{ int a[]=new int[5]; a[5]=30/0; } catch(ArithmeticException e) { System.out.println("task1 is completed"); } catch(ArrayIndexOutOfBoundsException e) { System.out.println("task 2 completed"); } catch(Exception e) { System.out.println("common task completed"); } System.out.println("rest of the code..."); } } </pre>	Remember	CO3	CACS552.08

2	<p>Trace the program and find out the output</p> <pre> public class Test { public static void aMethod() throws Exception { try { throw new Exception(); } finally { System.out.print("finally "); } } public static void main(String args[]) { try { aMethod(); } catch (Exception e) { System.out.print("exception "); } System.out.print("finished"); } } </pre>	Understand	CO3	CAC552.08
3	<p>Identify the output of the following program</p> <pre> class s1 implements Runnable { int x = 0, y = 0; int addX() { x++; return x; } int addY() { y++; return y; } public void run() { for(int i = 0; i < 10; i++) System.out.println(addX() + " " + addY()); } public static void main(String args[]) { s1 run2 = new s1(); Thread t1 = new Thread(run1); Thread t2 = new Thread(run2); t1.start(); t2.start(); } } </pre>	Understand	CO3	CAC552.11

4	<p>Interpret the output of following program</p> <pre> class Exceptions { public static void main(String[] args) { String languages[] = { "C", "C++", "Java", "Perl", "Python" }; try { for (int c = 1; c <= 5; c++) { System.out.println(languages[c]); } } catch (Exception e) { System.out.println(e); } } } </pre>	Apply	CO3	CAC552.09
5	<p>Identify the output of the below program</p> <pre> class Allocate { public static void main(String[] args) { try { long data[] = new long[1000000000]; } catch (Exception e) { System.out.println(e); } Finally { System.out.println("finally block will execute always."); } } } </pre>	Remember	CO3	CAC552.09
6	<p>Identify the output of the program</p> <pre> class MyThread extends Thread { public static void main(String [] args) { MyThread t = new MyThread(); Thread x = new Thread(t); x.start(); } public void run() { for(int i = 0; i < 3; ++i) { System.out.print(i + " .."); } } } </pre>	Remember	CO3	CAC552.10

7	<p>Trace the output of the program</p> <pre>class MyThread extends Thread { MyThread() { System.out.print(" MyThread"); } public void run() { System.out.print(" bar"); } public void run(String s) { System.out.println(" baz"); } } public class TestThreads { public static void main (String [] args) { Thread t = new MyThread() { public void run() { System.out.println(" foo"); } } } t.start(); }</pre>	Understand	CO3	CAC552.10
---	---	------------	-----	-----------

8	<p>Identify the output of the program</p> <pre> class MultithreadingDemo implements Runnable { public void run() { try { // Displaying the thread that is running System.out.println ("Thread " + Thread.currentThread().getId() + " is running"); } catch (Exception e) { // Throwing an exception System.out.println ("Exception is caught"); } } } // Main Class class Multithrea d { public static void main(String[] args) { int n = 8; // Number of threads for (int i=0; i<8; i++) { Thread object = new Thread(new MultithreadingDemo()); object.start(); } } } </pre>	Remember	CO3	CAC552.10
9	<p>Identify the output of the program</p> <pre> class implements Runnable { int x, y; public void run() { for(int i = 0; i < 1000; i++) synchronized(this) { x = 12; y = 12; } System.out.print(x + " " + y + " "); } public static void main(String args[]) { s run = new s(); Thread t1=new Thread(run); Thread t2=new Thread(run); t1.start(); t2.start(); } } </pre>	Remember	CO3	CAC552.10

UNIT-IV				
INTERFACES AND PACKAGES				
Part – A (Short Answer Questions)				
S. No	Questions	Blooms Taxonomy Level	Course Outcomes	Course Learning Outcomes
1	Define interface with an example.	Remember	CO4	CACS552.12
2	State the implementation of an interface in Java.	Remember	CO4	CACS552.12
3	Distinguish between the classes and interfaces.	Understand	CO4	CACS552.12
4	Define abstract method in java.	Remember	CO4	CACS552.12
5	Describe interface inheritance.	Understand	CO4	CACS552.12
6	Mention the use of java interface with an example.	Remember	CO4	CACS552.12
7	List out the types in inheritances in java.	Remember	CO4	CACS552.12
8	Define marker interface with an example.	Remember	CO4	CACS552.12
9	Define package with syntax.	Understand	CO4	CACS552.13
10	List out the types of packages in java.	Remember	CO4	CACS552.13
11	With syntax, write an example program of interfaces.	Understand	CO4	CACS552.12
12	List out advantages of packages in java.	Remember	CO4	CACS552.13
13	Define base class for all classes.	Understand	CO4	CACS552.13
14	Which package is always imported by default in java.	Remember	CO4	CACS552.13
15	Distinguish between class and a public class.	Remember	CO4	CACS552.13
16	Define abstract class with an example.	Remember	CO4	CACS552.12
17	What is classpath in java, explain with an example.	Understand	CO4	CACS552.13
Part – B (Long Answer Questions)				
1	Distinguish between abstract class and interface in detail.	Understand	CO4	CACS552.12
2	Explain different types of access specifiers used in java with an example.	Remember	CO4	CACS552.12
3	Describe the importance of interfaces by using an example	Remember	CO4	CACS552.12
4	Differentiate between classes and interfaces in java.	Understand	CO4	CACS552.12
5	Explain about importing packages.	Understand	CO4	CACS552.13
6	What is a classpath. Explain in detail with an example.	Understand	CO4	CACS552.13
7	Demonstrate how many ways packages can be imported.	Understand	CO4	CACS552.13
8	Explain how a class can extend an interface. Give an example.	Remember	CO4	CACS552.12
9	Discuss the advantage of using interface in Java.	Understand	CO4	CACS552.12
10	Explain about interface with an example.	Understand	CO4	CACS552.12
11	Can we have two public classes in a Java file? Explain with an example.	Remember	CO4	CACS552.13
12	Discuss in detail creating and importing package in java.	Understand	CO4	CACS552.13
13	Explain the process of creating a package with an example.	Understand	CO4	CACS552.13
14	Examine different ways to extending interfaces with an example.	Remember	CO4	CACS552.12

15	Can classes and interfaces are related to each other. Explain its differences	Understand	CO4	CACS552.12
16	Which keyword java compiler add before interface fields and methods	Understand	CO4	CACS552.12
17	Demonstrate compile-time error with an example.	Understand	CO4	CACS552.12
18	Distinguish between class, abstract class and interface.	Remember	CO4	CACS552.12
19	Can abstract class implements interface in Java? does they require to implement all methods?	Remember	CO4	CACS552.12
Part – C (Problem Solving and Critical Thinking Questions)				
1	<p>Identify the output of the following program.</p> <pre> interface Sample { int x=12; void show(); default void display() { System.out.println("default method of interface"); } Static void print(String str) { System.out.println("Static method of interface:"+str); } } </pre>	Understand	CO4	CACS552.12
2	<p>Predict the output of the following program</p> <pre> interface calculate { void cal(int item); } class display implements calculate { int x; public void cal(int item) { x = item * item; } } class interfaces { public static void main(String args[]) { display arr = new display; arr.x = 0; arr.cal(2); System.out.print(arr.x); } } </pre>	Apply	CO4	CACS552.12
3	<p>Discuss the output of the following program?</p> <pre> interface MyInterface { public void method1(); public void method2(); } class XYZ implements MyInterface { public void method1() { </pre>	Understand	CO4	CACS552.12

	<pre> System.out.println("implementation of method1"); } public void method2() { System.out.println("implementation of method2"); } public static void main(String arg[]) { MyInterface obj = new XYZ(); obj. method1(); }} </pre>			
4	<p>Identify the output of the following program.</p> <pre> interface Inf1 { public void method1(); } interface Inf2 extends Inf1 { public void method2(); } public class Demo implements Inf2{ public void method1(){ System.out.println("method1"); } public void method2(){ System.out.println("method2"); } public static void main(String args[]) { Inf2 obj = new Demo(); obj.method2(); } } </pre>	Understand	CO4	CACS552.12
5	<p>Identify the output of the following program.</p> <pre> Interface A { int x=10; } interface B { int x=100; } class Hello implements A,B { public static void Main(String args[]) { /* reference to x is ambiguous both variables are x * so we are using interface name to resolve the * variable */ System.out.println(x); System.out.println(A.x); System.out.println(B.x); } } </pre>	Understand	CO4	CACS552.12
6	<p>Identify the output of the following</p> <pre> interface IShape { void f1(); void f2(); } </pre>	Understand	CO4	CACS552.12

	<pre> void f3(); } class Circle implements IShape { public void f1() { } } </pre>			
--	---	--	--	--

UNIT-V

FILES AND CONNECTING TO DATABASE

Part - A (Short Answer Questions)

1	Define the term stream with an example.	Remember	CO5	CACS552.15
2	Define I/O stream with an example.	Remember	CO5	CACS552.16
3	List out various types of streams in java.	Understand	CO5	CACS552.16
4	Define the syntax to create I/O streams.	Remember	CO5	CACS552.16
5	Describe the method used to read the data through keyboard.	Understand	CO5	CACS552.15
6	Define Scanner class with an example.	Remember	CO5	CACS552.15
7	Describe the use of method ExecuteUpdate() in database connectivity.	Understand	CO5	CACS552.18
8	Define the package for JDBC.	Remember	CO5	CACS552.17
9	List out various steps involved in database connection.	Remember	CO5	CACS552.17
10	What is JDBC Driver? List the types of JDBC drivers.	Understand	CO5	CACS552.17
11	Define the importance of FileInputStream and FileOutputStream.	Remember	CO5	CACS552.16
12	Explain the statement "System.out.println()"	Understand	CO5	CACS552.16
13	List out various types of JDBC Drivers.	Remember	CO5	CACS552.15
14	Define the term byte stream.	Remember	CO5	CACS552.17
15	Define standard streams in java.	Understand	CO5	CACS552.16
16	Write a short note on character stream.	Remember	CO5	CACS552.16

Part - B (Long Answer Questions)

1	Write a java program to insert record in a table.	Understand	CO5	CACS552.18
2	Write a java program to display records of a table.	Remember	CO5	CACS552.18
3	Explain InputStream hierarchy with a neat sketch.	Understand	CO5	CACS552.16
4	Illustrate various steps for connecting to database with a scriplet.	Understand	CO5	CACS552.17
5	How Statement.executeUpdate() is used to update the table records. Explain with an example.	Understand	CO5	CACS552.18
6	Explain OutputStream hierarchy with a neat sketch.	Remember	CO5	CACS552.16
7	Write a java program to update records in a table.	Understand	CO5	CACS552.19
8	Explain briefly the use of JDBC-ODBC driver with a neat sketch.	Understand	CO5	CACS552.17
9	Explain in detail various JDBC statements.	Understand	CO5	CACS552.17
10	Write a program to update the salary Rs.10000/- for an employee name like "sita" using prepared statement.	Understand	CO5	CACS552.18
11	What are the differences between Statement and PreparedStatement interface?	Understand	CO5	CACS552.17

12	With an example, write JDBC prepared statement with ResultSet.	Understand	CO5	CACS552.17
13	Write a program to execute and read select queries by using JDBC in java.	Understand	CO5	CACS552.18
14	Illustrate a JDBC application for querying the database and processing the results.	Remember	CO5	CACS552.17
15	Explain execute query with an example.	Understand	CO5	CACS552.18
16	Distinguish between a)InputStream and Reader classes b)OutputStream and Writer Classes.	Understand	CO5	CACS552.16
17	Explain different types of JDBC drivers with a neat sketch.	Understand	CO5	CACS552.18
18	What are the steps to connect to the database in java?	Remember	CO5	CACS552.18
19	What are the JDBC API components?	Understand	CO5	CACS552.18
20	Distinguish between execute, executeQuery, and executeUpdate?	Remember	CO5	CACS552.17
Part – C (Problem Solving and Critical Thinking Questions)				
1	Identify be the output of the program? import java.io.*; class filesinputoutput { public static void main(String args[]) { InputStream obj = new FileInputStream("inputoutput.java"); System.out.print(obj.available()); } }	Remember	CO5	CACS552.16
2	Examine the following program and find the output public static void main(String[] args) { String name = null; File file = new File("/folder", name); System.out.print(file.exists()); }	Remember	CO5	CACS552.18
3	Interpret the output of the program. import java.io.*; public class filesinputoutput { public static void main(String[] args) { String obj = "abc"; byte b[] = obj.getBytes(); ByteArrayInputStream obj1 = new ByteArrayInputStream(b); for (int i = 0; i < 2; ++ i) { int c; while((c = obj1.read()) != -1) { if(i == 0) { System.out.print(Character.toUpperCase((char)c)); obj2.write(1); } } System.out.print(obj2); } }}	Evaluate	CO5	CACS552.16

4	<p>Predict the output of the program.</p> <pre> import java.io.*; class Chararrayinput { public static void main(String[] args) { String obj = "abcdef"; int length = obj.length(); char c[] = new char[length]; obj.getChars(0, length, c, 0); CharArrayReader input1 = new CharArrayReader(c); CharArrayReader input2 = new CharArrayReader(c, 0,3); int i; try { while((i = input2.read()) != -1) { System.out.print((char)i); } } catch (IOException e) { e.printStackTrace(); } } </pre>	Understand	CO5	CAC552.16
5	<p>Examine the following code and define the meaning.</p> <pre> import java.util.*; String URL = "jdbc:oracle:thin:@amrood:1521:EMP"; Properties info = new Properties(); info.put("user", "username"); info.put("password", "password"); Connection conn = DriverManager.getConnection(URL, info); </pre>	Remember	CO5	CAC552.18
6	<p>Describe about the following code.</p> <pre> static final String USER = "username"; static final String PASS = "password"; System.out.println("Connecting to database..."); conn = DriverManager.getConnection(DB_URL,USER,PASS); System.out.println("Creating statement..."); stmt = conn.createStatement(); String sql; sql = "SELECT id, first, last, age FROM Employees"; ResultSet rs = stmt.executeQuery(sql); </pre>	Understand	CO5	CAC552.18

Prepared by,
Mr. G Chandra Sekhar, Assistant Professor, CSE, Dept.

HOD, ECE