INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad -500 043

ELECTRONICS AND COMMUNICATION ENGINEERING

TUTORIAL QUESTION BANK

Course Name	**JAVA PROGRAMMING**
Course Code	ACS552
Class	B. Tech VI Semester
Branch	Electronics And Communication Engineering
Year	2018 – 2019
Course Faculty	Ms. Y Harika Devi, Assistant Professor Mr. S Laxman Kumar, Assistant Professor Mr. N V Krishna Rao, Assistant Professor Ms. G Geetha, Assistant Professor Mr. Santosh Patel, Assistant Professor

COURSE OBJECTIVES:

The course should enable the students to:

I	Understand the basic object oriented programming concepts and apply them in problem solving.
II	Illustrate inheritance concepts for reusing the program.
III	Demonstrate on the multi-tasking is performed by using multiple threads.
IV	Develop data-centric applications using JDBC.
V	Understand the basics of java file management

COURSE LEARNING OUTCOMES:

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

CACS552.01	Use object oriented programming concepts to solve real world problems.
CACS552.02	Explain the concept of class and objects with access control to represent real world entities.
CACS552.03	Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors.
CACS552.04	Describe the concept of operators and variables, arrays, parameter passing.
CACS552.05	Use overloading methodology on methods and constructors to develop application programs.
CACS552.06	Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords.
CACS552.07	Use dynamic and static polymorphism to process objects depending on their class.
CACS552.08	Understand the impact of exception handling to avoid abnormal termination of program using checked and unchecked exceptions.
CACS552.09	Demonstrate the user defined exceptions by exception handling keywords (try, catch, throw, throws and finally).
CACS552.10	Use multithreading concepts to develop inter process communication.
CACS552.11	Understand the use of interrupting threads in the real world.
CACS552.12	Describe the concept of interface and abstract classes to define generic classes.
CACS552.13	Illustrate different techniques on creating and accessing packages (fully qualified name and import statements).
CACS552.14	Demonstrate the import statement usage and built-in packages.

CACS552.15	Understand text, byte, and character input/output streams.
CACS552.16	Understand and implement concepts on file streams and operations in java programming for a given application programs.
CACS552.17	Describe the backend connectivity process in java program by using JDBC drivers.
CACS552.18	Develop java application to interact with database by using relevant software component (JDBC Driver).
CACS552.19	Demonstrate the use of programming in the real world.
CACS552.20	Posses the knowledge and skills for employability and to succeed in national and international level competitive exams.

TUTORIAL QUESTION BANK

UNIT – I

OOPS CONCEPTS AND JAVA PROGRAMMING

PART – A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Learning Outcomes
1	State importance of Object Oriented Programming.	Remember	CACS552.01
2	Distinguish between procedural language and OOP's.	Understand	CACS552.01
3	Define class with an example.	Remember	CACS552.01
4	Define Inheritance and list out inheritance types.	Understand	CACS552.01
5	Define Polymorphism with an example.	Remember	CACS552.01
6	Distinguish between C, C++ and java.	Remember	CACS552.01
7	List out java programming buzzwords.	Remember	CACS552.01
8	Describe history of java.	Understand	CACS552.01
9	List out different data types used in java.	Remember	CACS552.03
10	Define object with example.	Remember	CACS552.01
11	Describe scope and life time of variables.	Understand	CACS552.04
12	List and describe different types of operators.	Remember	CACS552.01
13	Define array with an example.	Understand	CACS552.02
14	Define expressions with an example.	Remember	CACS552.01
15	Define enumerated types with an example.	Remember	CACS552.01
16	List out different types of control flow statements.	Understand	CACS552.03
17	List out advantages of inheritance.	Remember	CACS552.04
18	Distinguish between constructor and method.	Understand	CACS552.05
19	Define data abstraction with an example.	Remember	CACS552.01
20	Distinguish type conversion and type casting.	Understand	CACS552.01

Part - B (Long Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Learning Outcomes
1	Explain about OOP's concepts with an example.	Understand	CACS552.01
2	Explain briefly about the features (buzzwords) of Java.	Understand	CACS552.01
3	Java is a pure object oriented programming language. Justify.	Remember	CACS552.01
4	Explain in detail about JVM architecture.	Understand	CACS552.01
5	Explain the importance of this keyword with an example.	Understand	CACS552.04
6	Interpret method overloading with an example.	Understand	CACS552.04
7	Discuss about the constructor overloading with an example.	Understand	CACS552.04
8	Explain the concept of arrays with an example.	Understand	CACS552.03
9	Explain in detail about static block with an example.	Understand	CACS552.02
10	Discuss about various conditional statements in java with suitable examples.	Understand	CACS552.03
11	List out various types of variables in detail with an example.	Understand	CACS552.03
12	List out various ways for creation of object for a class.	Remember	CACS552.03
13	Describe method in detail with an example.	Understand	CACS552.03
14	Explain narrowing and widening in detail with an example.	Understand	CACS552.03
15	Describe about static variable with an example.	Remember	CACS552.04

16	Describe static method with an example.	Remember	CACS552.02
17	Interpret type conversion and casting with an example.	Understand	CACS552.03
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	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.pirnltn(“ y is :” + y); y=100; System.out.println(“ y is now : “ + y);
	}
	}
	}
		Evaluate	CACS552.03

3	Predict output of the program.
	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);
	}
	}
		Apply	CACS552.03

4	Explain the following code is valid or not. public String getDescription(Object obj)
	{
	return obj.toString;
	}
	public String getDescription(String obj)
	{
	return obj;
	}
	public void getDescription(String obj)
	{
	return obj;
	}
		Understand	CACS552.03

5	Predict the output of following program?
	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);
	}
	}
		Apply	CACS552.02

6	Identify output of the program?
	
	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);
	}
	}
		Remember	CACS552.03

7	Predict the following program output.
	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;
	}
	}
	}
		Apply	CACS552.04

8	Identify output of the program?
	class BitShift
	{
	public static void main(String [] args)
	{
	int x = 0x80000000;
	System.out.print(x + " and ");
	x = x >>> 31;
	System.out.println(x);
	}
	}
		Remember	CACS552.03

9	Predict the program and find out the output.
	class Equals
	{
	public static void main(String [] args)
	{
	int x = 100; double
	y = 100.1;
	boolean b = (x = y);
	System.out.println(b);
	}
	}
		Apply	CACS552.05

UNIT – II

INHERITANCE

Part – A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Learning Outcomes
1	Write a short notes on inheritance.	Remember	CACS552.06
2	List various types of inheritances in java.	Remember	CACS552.06
3	Define the term static binding.	Remember	CACS552.06
4	Define run time polymorphism.	Remember	CACS552.07
5	Describe abstract class.	Understand	CACS552.07
6	Interpret various member access rules in java.	Understand	CACS552.07
7	Define method overriding.	Remember	CACS552.06
8	Define polymorphism method overloading.	Remember	CACS552.07
9	Define and use final keyword to prevent inheritance using example	Remember	CACS552.06
10	Discuss the syntax of Inheritance	Understand	CACS552.06
11	Define compile time polymorphism.	Understand	CACS552.07
12	Define and use final keyword to prevent overriding using example	Remember	CACS552.06
13	Discuss the forms of inheritance	Understand	CACS552.06
14	Describe multiple inheritance is not supported by Java.	Understand	CACS552.06
15	Differentiate Inheritance and Encapsulation	Understand	CACS552.06
16	Define dynamic binding with an example.	Remember	CACS552.07

Part - B (Long Answer Questions)

1	Differentiate “this” and “super” keywords usage in java.	Understand	CACS552.06
2	List different types of inheritances in java with example.	Remember	CACS552.06
3	Discuss various methods of Object class.	Understand	CACS552.06
4	Illustrate the Use of “Super” keyword in method overriding with example.	Understand	CACS552.06
5	Compare and Contrast interfaces and Abstract classes.	Understand	CACS552.06
6	Demonstrate dynamic binding with an example.	Understand	CACS552.06
7	List out the some of the standard overloaded methods in java.	Remember	CACS552.06
8	Describe Abstraction in java using abstract class with an example.	Remember	CACS552.06
9	Illustrate what happens if the parent and the child class have a field with same identifier.	Understand	CACS552.07
10	Define multiple inheritances with suitable example.	Remember	CACS552.06
12	Compare and contrast overloading and overriding methods.	Understand	CACS552.07

14	State which method hides a method in the superclass.	Remember	CACS552.06
15	Discuss the importance of final keyword in java with a program.	Understand	CACS552.06
16	State benefits of inheritance with an example.	Remember	CACS552.06
17	Show hierarchal abstractions.	Understand	CACS552.06
18	State concepts of inheritance.	Remember	CACS552.06
	Part – C (Problem Solving and Critical Thinking)		

1	Interpret the program and give output.
	
	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();
	}}
		Evaluate	CACS552.06

2	Identify the output of the following program.
	class A{
	int i = 10;
	}
	class B extends A{
	int i = 20;}
	public class MainClass
	{
	public static void main(String[] args){
	A a = new B();
	System.out.println(a.i);}
	}
		Remember	CACS552.07

3	Predict output of the program?
	class A
	{
	public A()
	{
	System.out.println("NewA");
	}}
	class B extends A
	{public B()
	{
	super();
	System.out.println("New B");
	}
	}
		Apply	CACS552.06

4	Discuss the output of the following program?
	class X
	{
	public X(int i)
	{
	System.out.println(1);
	}
	}
	class Y extends X
	{
	public Y()
	{
	System.out.println(2);
	}
	}
		Understand	CACS552.07

5	Predict the program
	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();
	}
	}
		Apply	CACS552.06

6	Discuss the output of the program?
	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);
	}
	}
		Understand	CACS552.06

7	Interpret and find out the output of the program? 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);
	}
	}
		Evaluate	CACS552.06

8	Illustrate a java program to create an abstract class named Shape that contains two integers and an empty method named print Area().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 print Area () that prints the area of the given shape.	Remember	CACS552.07
	UNIT-III		
	EXCEPTION HANDLING AND MULTITHREADING		
	Part - A (Short Answer Questions)		
S. No	**Questions**	**Blooms Taxonomy Level**	**Course Learning Outcomes**
1	Define the term Exception.	Remember	CACS552.08
2	Distinguish between exception and error.	Understand	CACS552.08
3	Describe the benefits of exception handling.	Remember	CACS552.09
4	State the classification of exceptions.	Remember	CACS552.08
5	Define checked exceptions.	Remember	CACS552.08
6	State the use of try and catch blocks.	Remember	CACS552.08
7	Describe built in exception.	Understand	CACS552.09
8	List Exception handling keyword.	Remember	CACS552.09
9	Define unchecked exceptions.	Understand	CACS552.09
10	Define nested try catch block with an example.	Remember	CACS552.08
11	Define thread in java. List the various ways of creating thread.	Remember	CACS552.10
12	Describe the various states of threads.	Understand	CACS552.10
13	List the different ways to create a thread.	Remember	CACS552.10
14	Differentiate throw and finally.	Understand	CACS552.10

15	Define inter-thread communication.	Remember	CACS552.10
16	State about the alive() and join() method	Remember	CACS552.10
17	Interpret the different thread priorities	Understand	CACS552.10
18	Distinguish between throw and throws.	Understand	CACS552.10
19	Define wait() state of the thread	Remember	CACS552.11
20	Describe about “thread class implements Runnable interface”	Remember	CACS552.11
21	Compare and contrast between process and thread.		
	Part – B (Long Answer Questions)		
1	Explain briefly about exception handling mechanism with suitable Examples.	Understand	CACS552.08
2	Describe try, catch , and finally keywords with an example	Remember	CACS552.08
3	Illustrate use of throws keyword with a program	Understand	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	CACS552.08
5	Differentiate between checked and unchecked exceptions.	Understand	CACS552.09
6	Exemplify the different types of exception.	Understand	CACS552.08
7	Illustrate built in exceptions with suitable example.	Understand	CACS552.09
8	Explain throwing of user defined exception with example	Understand	CACS552.09
9	Describe the producer consumer problem with an example	Remember	CACS552.11
10	Explain with an example how java performs thread synchronization.	Understand	CACS552.10
11	Differentiate multiprocessing and multithreading with a program.	Understand	CACS552.10
12	Explain briefly about the life cycle of a thread with an example.	Understand	CACS552.11
13	Interpret various methods of thread class.	Understand	CACS552.11
14	Describe a java program using thread priorities.	Remember	CACS552.10
15	Explain Daemon threads with an example.	Understand	CACS552.10
16	Exemplify the behavior of thread using thread class methods.	Understand	CACS552.11
17	Illustrate the process of creating thread by implementing Runnable interface.	Remember	CACS552.11
	Part – C (Problem Solving and Critical Thinking Questions)		
1	Identify the output of program	Remember	CACS552.08

```
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...");
         }
}
```

2	Trace the program and find out the output
	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");
	}
	}
		Understand	CACS552.08

3	Discuss the output of the following program
	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();
	Threadt1 = new Thread(run1);
	Thread t2 =new Thread(run2);
	t1.start();
	t2.start();
	}
		Understand	CACS552.11

4	Interpret the output of following program 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);
	}
	}
	}
		Apply	CACS552.09

5	Identify the output of the below program
	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.");
	}
	}
	}
		Remember	CACS552.09

6	Identify the output of the program
	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 + "..");
	}
	}
	}
		Remember	CACS552.10

7	Trace the output of the program
	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();
	}
	}
		Understand	CACS552.10

8	Identify the output of the program
	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 Multithread
	{
	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());
		Remember	CACS552.10

	object.start();
	}
	}
	}		

9	Identify the output of the program
	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();
	}
	}
		Remember	CACS552.10

UNIT-IV

INTERFACES AND PACKAGES

Part – A (Short Answer Questions)

S. No	Questions	Blooms Taxonomy Level	Course Learning Outcomes
1	Define interface with an example.	Remember	CACS552.12
2	State implementation of an interface in Java.	Remember	CACS552.12
3	Define relationship between classes and interfaces.	Understand	CACS552.12
4	Define abstract method in java.	Remember	CACS552.12
5	Describe interface inheritance.	Understand	CACS552.12
6	Mention use of java interface with an example.	Remember	CACS552.12
7	List out types in inheritances in java.	Remember	CACS552.12
8	Define marker interface with an example.	Remember	CACS552.12
9	Define package with syntax.	Understand	CACS552.13
10	List out types of packages in java.	Remember	CACS552.13
11	Write simple java code for interface.	Understand	CACS552.12
12	List advantages of packages in java.	Remember	CACS552.13
13	Define base class for all classes.	Understand	CACS552.13
14	Which package is always imported by default in java.	Remember	CACS552.13
15	Distinguish class and public class.	Remember	CACS552.13
16	Define abstract class with an example.	Remember	CACS552.12
17	Define classpath in java.	Understand	CACS552.13

Part – B (Long Answer Questions)

1	Distinguish abstract class and interface in detail.	Understand	CACS552.12
2	Explain different access specifiers with an example.	Remember	CACS552.12
3	Describe interfaces using with an example	Remember	CACS552.12
4	Differentiate classes and interfaces inf java.	Understand	CACS552.12
5	Explain about importing packages.	Understand	CACS552.13
6	What is a classpath. Explain in detail	Understand	CACS552.13

7	Demonstrate in how many ways packages can be imported.	Understand	CACS552.13
8	Examine can a class extend an interface. Give an example	Remember	CACS552.12
9	Discuss the advantage of using interface in Java.	Understand	CACS552.12
10	Explain about interface with an example.	Understand	CACS552.12
11	Can we have two public classes in a Java file. Explain with an example.	Understand	CACS552.13
12	Discuss in detail creating and importing package in java.	Understand	CACS552.13
13	Explain package creation with an example.	Understand	CACS552.13
14	Examine different ways to extending interfaces with an example.	Remember	CACS552.12
15	Differentiate between class and interface.	Understand	CACS552.12
16	Illustrate interfaces with two examples in java.	Understand	CACS552.12
17	Demonstrate compile-time error with an example.	Understand	CACS552.12
18	Distinguish abstract class and interface with syntax.	Remember	CACS552.12
Part – C (Problem Solving and Critical Thinking Questions)			

1	Analyze the output
	class A implements B
	{
	public int methodB(int i)
	{
	return i =+ i * i;
	}
	}
	
	interface B
	{
	int methodB(int i);
	}
	
	public class MainClass
	{
	public static void main(String[] args)
	{
	B b = new A();
	System.out.println(b.methodB(2));
	}
	}
		Understand	CACS552.12

2	Identify the output of the following program.
	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);
	}
	}
		Understand	CACS552.12

3	Predict output of the program?
	interface A
	{
	System.out.println("Interface A");{
	}
	static
	{
	System.out.println("Interface A");
	}
	}
		Understand	CACS552.12

4	Discuss the output of the following program?
	interface MyInterface
	{
	public void method1();
	public void method2();
	}
	class XYZ implements MyInterface
	{
	public void method1()
	{
	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();
	}
	}
		Understand	CACS552.12

UNIT-V

FILES AND CONNECTING TO DATABASE

Part - A (Short Answer Questions)

1	Define the term stream with an example.	Remember	CACS552.15
2	Define I/O stream with an example.	Remember	CACS552.16
3	List out various types of streams in java.	Understand	CACS552.16
4	Define syntax to create I/O streams.	Remember	CACS552.16
5	Describe the method used to read the data through keyboard.	Understand	CACS552.15
6	Define Scanner class with an example.	Remember	CACS552.15
7	Describe the use of method ExecuteUpdate() in database connectivity.	Understand	CACS552.18
8	Define the package for JDBC.	Remember	CACS552.17
9	List out the steps in database connection.	Remember	CACS552.17
10	Describe DDL, DML commands in JDBC.	Understand	CACS552.17
11	Define Fileinputstream and Fileoutputstream.	Remember	CACS552.16
12	Define System.out.println().	Understand	CACS552.16
13	List out various types of JDBC Drivers.	Remember	CACS552.15
14	Define the term byte stream.	Remember	CACS552.17
15	Define standard streams in java.	Understand	CACS552.16
16	Write a short notes on character stream.	Remember	CACS552.16

Part - B (Long Answer Questions)

1	Write a java program to insert record in a table.	Understand	CACS552.19
2	Write a program to display records of a table.	Remember	CACS552.19
3	Explain inputstream Hierarchy with a neat sketch.	Understand	CACS552.16
4	Illustrate steps for connecting to database with a scriplet.	Understand	CACS552.17

5	How Statement.executeUpdate() is used to update the table records. Explain with an example.	Understand	CACS552.18
6	Explain outputstream Hierarchy with a neat sketch.	Remember	CACS552.16
7	Write a java program to update records in a table.	Understand	CACS552.19
8	Explain JDBC-ODBC driver.	Understand	CACS552.17
9	Explain statement and resultset in JDBC with syntax.	Understand	CACS552.17
10	Explain the program to update the salary Rs.10000/- for an employee name like “sita” using prepared statement.	Understand	CACS552.18
11	Explain about hierarchy of classes with io streams.	Understand	CACS552.15
12	Write an example for JDBC prepared statement with ResultSet.	Understand	CACS552.17
13	Explain the program to execute and read select queries using JDBC in java.	Understand	CACS552.18
14	Illustrate a JDBC application for querying the database and processing the results.	Remember	CACS552.17
15	Explain execute query with an example.	Understand	CACS552.19
16	Distinguish between a)InputStream and Reader classes b)OutputStream and Writer Classes.	Understand	CACS552.16
17	Explain different types of JDBC drivers with a neat sketch.	Understand	CACS552.18
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	CACS552.16

2	Examine the following program and find the
	public static void main(String[] args){
	String name = null;
	File file = new File("/folder", name);
	System.out.print(file.exists());
	}
		Remember	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	CACS552.16

4	Predict the output of the program.
	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();
	}
	}
	}
		Understand	CACS552.16

5	Examine the following code and define the meaning.
	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);
		Remember	CACS552.18

6	Describe about the following code.
	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);
		Understand	CACS552.19

Prepared by,
Ms. Y Harika Devi, Assistant Professor, CSE Dept.

HOD, ECE