Name: Enrolment No:	UPES UNIVERSITY WITH A PURPOSE

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
Online End Semester Examination, December 2020

Course: JAVA SE Fundamentals **Semester: III**
Program: BCA – IoT, BCA – BFSI **Time: 03 hrs.**
Course Code: CSBC 2012 **Max. Marks: 100**

SECTION A

1. **Each Question will carry 5 Marks**
2. **Instruction: Select the correct answer(s)**

S. No.	**Question**	**CO**

Q 1	State the error (if any) and give the output (if no error found).
	abstract class demo
	{
	public int a;
	demo()
	{
	a = 10;
	}
	
	abstract public void set();
	abstract final public void get();
	}
	
	class Test extends demo
	{
	public void set(int a)
	{
	this.a = a;
	}
	final public void get()
	{
	System.out.println("a = " + a);
	}
	public static void main(String[] args)
	{
	Test obj = new Test();
	obj.set(20);
	obj.get();
	}
		CO3

	}	
Q2	Which statements are true? a. There are eight primitive types built into the Java programming language. b. byte, short, char, and long are the four integral primitive data types in the Java programming language. c. A Boolean type variable holds true, false, and nil. d. short Long = 10; is a valid statement that adheres to the variable declaration and initialization syntax.	**CO1**

Q3	What should be the execution order, if a class has a method, static block, instance block, and constructor, as shown below?
	public class First_C {
	public void myMethod()
	{
	System.out.println("Method");
	}
	
	{
	System.out.println(" Instance Block");
	}
	
	public void First_C()
	{
	System.out.println("Constructor ");
	}
	static {
	System.out.println("static block");
	}
	public static void main(String[] args) {
	First_C c = **new** First_C();
	c.First_C();
	c.myMethod();
	}
	}
	a. Instance block, method, static block, and constructor
	b. Method, constructor, instance block, and static block
	c. Static block, method, instance block, and constructor
	d. Static block, instance block, constructor, and method
		CO3

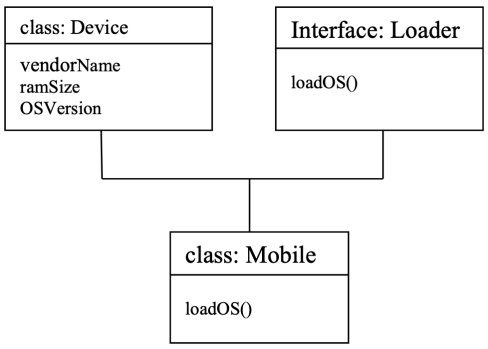
Q4	What will be the output of the following JAVA program?
	class Base extends Exception {}
	class Derived extends Base {}
	public class Main {
	  public static void main(String args[]) {
	  // some other stuff
	  try {
	  // Some monitored code
	  throw new Derived();
	  }
	  catch(Base b) {
	  System.out.println("Caught base class exception");
	  }
	  catch(Derived d) {
	  System.out.println("Caught derived class exception");
	  }
	  }
	}
	
	a. Caught base class exception
	b. Caught derived class exception
	c. Compiler Error because derived is not throwable
	d. Compiler Error because base class exception is caught before derived class
		CO4

Q5	Name the methods from wrapper class for following task i) To convert string objects to primitive int. ii) To convert integer object to string object.	**CO2**

Q6	State the error (if any) and give the output (if no error found).
	
	class Test{
	  public void display(int x, double y){
	  System.out.println(x+y);
	  }
	  public double display(int p, double q){
	  return (p+q);
	  }
	  public static void main(String args[]){
	  Test test = new Test();
	  test.display(4, 5.0);
	  System.out.println(test.display(4, 5.0));
	  }
	}
		CO2

SECTION B

1. **Each question will carry 10 marks**
2. **Instruction: Write short / brief notes**

S. No.	**Question**	**CO**
Q 7	(a) Describe the following String methods with example: (6) i. replace() ii. compareTo() iii. charAt() iv. substring() (b) What is the significance of Java's byte code? (4)	**CO1**
Q 8	(a) Discuss various access modifiers available in Java? How access modifier effect the visibility of a member in different access locations? Explain with an example. (6) (b) What is a constructor in Java? How many types of constructors are there in Java? Explain with examples. (4)	**CO2**
Q 9	Implement following inheritance:  Display details of devices from loadOS() method of class Mobile.	**CO3**
Q 10	Comment -"Error and Exception are two different terms". Create a user defined exception named BallotCount to count no. of votes in each party. If NA is selected throw the IllegalArgumentException.	**CO4**
Q 11	(a) With suitable code segment illustrate the use of 'this' keyword in Java. (5) (b) Differentiate between method overloading and method overriding. (5) OR (a) State five similarities between Interfaces and Classes. (5) (b) Describe the uses of final and super keywords with respect to inheritance. (5)	**CO3**

Section C

1. **Each Question carries 20 Marks.**
2. **Instruction: Write long answer.**

Q12	A bank maintains two kinds of accounts – Savings account and Current account. The savings account provides simple interest, deposit and withdrawal facilities. The current account only provides deposit and withdrawal facilities. Current account holders should also maintain minimum balance. If balance falls below minimum level, a service charge is imposed. Create an abstract class Account that stores customer name, account number type of account and abstract methods. From this derive the classes Curr_Account (double balance, double min_bal, double serviceCharge / penalty) and and Sav_Account (double balance). Include the necessary methods in order to achieve the following: a. Define parameterized constructor in a class hierarchy. b. Allow deposit and update the balance. c. Display the balance. d. Compute interest and add to balance. e. Permit withdrawal and update the balance (check for minimum balance). f. Apply polymorphism if required for methods in class hierarchy. g. Create an array of super class / object and populate with subclass objects and call the overridden / object methods. h. Write a test program to demonstrate the above said implementations. OR Define a interface EMPInterface (void displayEMP(), void giveBonus (double amount)). Define an abstract class Employee (empID, fName, lName, salary). Define a concrete class Manager (noOfOtockOptions), subclass of Employee and define interface methods. a. Define appropriate constructors in a class hierarchy. b. Ensure the bonus amount should not be negative and zero using exception handling mechanism (use throws and throw clauses of exception handling). c. Create array of interface reference variables and populate with manager objects. d. Write a test program to implement the above said requirements of interface implementation and exception handling.	**CO3**