Nepal College of Information Technology
Balkumari, Lalitpur

Tutorial-2

Subject: Programming Technology
Program: BECE IV Deadline: **2077 Chaitra 20**

Chapter 4,5,6 and 7

Note: Submit the soft copy of your work via LMS or hard copy through Department of Computer Engineering, NCIT.

1. Explain the seven key pillars of ASP.net.
2. Clarify your understanding and uses of ASP.Net file types and ASP.net application directories.
3. What are the server controls in ASP.net? Explain the life cycle of asp.net page.
4. What is connection string? Explain the architecture of ADO.NET.
5. Write a simple asp.net program in order to demonstrate the connectivity to the database server (Use connection string).
6. What is JIT compilation? Differentiate JVM, JRE and JDK.
7. How error is differ from exception. Write a java program to illustrates the use of try, catch, throw, throws and finally keyword.
8. Write a Java program to demonstrate the concept of user defined exception taking any example scenario.
9. What is debugging? Explain different debugging techniques with suitable example section of code.
10. Explain each of the following with relevant example program: How Java programming provides the basic object oriented features like encapsulation, abstraction , polymorphism and inheritance.
11. What are the significance of using interface in Java? Explain with suitable example program.
12. Write a program to implement following functionality of Bank Account. Assume necessary member variable required. (**Consider exception handling too**)

    Class Name : *BankAccount*
    Member Methods:
    public BankAccount(long accountNumber) : use parameterized Constructor
    public void deposit(double amount) : deposit an amount
    public void withdraw(double amount) : Withdraw an amount (check balance before withdraw)
    public double getBalance() : Return an available balance
    public void transfer(double amount, BankAccount targetAccount) : Transfer amount to targetAccount
13. Make a Java application with these features:
    - An interface called **RegularPolygon** with two abstract methods: **getNumSides** and **getSideLength**.
    - A class **EquilateralTriangle** that implements the interface, has **getNumSides** return 3 and **getSideLength** return an instance variable that is set by the constructor.
    - A class **Square** that implements the interface, has **getNumSides** return 4 and **getSideLength** return an instance variable that is set by the constructor.

    13.1. Add a static **totalSides** method, that given a **RegularPolygon[]**, returns the sum of the number of sides of all the elements.
    13.2. Add two default methods:
        - **getPerimeter** (n * length, where n is the number of sides)
        - **getInteriorAngle** ($(n-2)\pi/n$ in radians)
    13.3. Make a few test cases.