BUBT
Committed to Academic Excellence

BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY

Assignment 1

Course Code: CSE 331

Course Title: Advanced Programming

Submitted to:	Submitted by:
Name: Md. Anwar Hussen Wadud Lecturer, Dept. of CSE at Bangladesh University of Business and Technology.	Name: Syeda Nowshin Ibnat ID: 17183103020 Intake :39 Section: 01 Program: B.Sc. in CSE Semester: Summer 2020

Date of Submission: 24.10.2020

Assignment - 1

Question: Some programmers prefer not to use protected access, because they believe it breaks the encapsulation of the superclass. Discuss the relative merits of using protected access vs. using private access in super classes.

Solution: Inherited Private instance variables are hidden in the subclass and are accessible only through the public or protected methods of the superclass. Using protected access enables the subclass to manipulate the protected members without using the access methods of the superclass. This makes the code more brittle, because changes to the superclass might require changes to the subclasses. If the superclass instance variable are private, the methods of the superclass must be used to access the data. This encapsulation makes the code easier to maintain, modify and debug.

Protected access

The methods or data members declared as protected are accessible within same package or sub classes in different package.

For example:

```
Package P1;

Public class A
{ protected void display()
{ System.out.println(" It's an protected access modifier."); }}

Package P2;
impont P1.*;
class B extends A
{ Public static void main(String args[])
{ B obj = new B();
  obj.display(); }}
```

Output:

It's a protected access modifier.

Private access

Any other class of same package will not be able to access these members.

For example:

```
Package P1;
class A {
private void display(){
 System.out.println(" It's a private access modifier."); }}
class B{
Public static void main(String args[]){
 A Object = new A();
  obj.display(); }}
```

Output:

error: display() has private access in A obj.display();

Ans: