**Student Name: M.Naeem Student ID: 16213**

**Degree: BS(CS) Semester: 2nd**

**Subject: Object Oriented Programming Instructor: Sir M.Ayub Khan**

**Final Term Exam Spring 2020 Date: 29/06/2020**

**Q NO 1: a. Why access modifiers are used in java, explain in detail Private and Defaul access modifiers?**

**b. Write a specific program of the above mentioned access modifiers in java.**

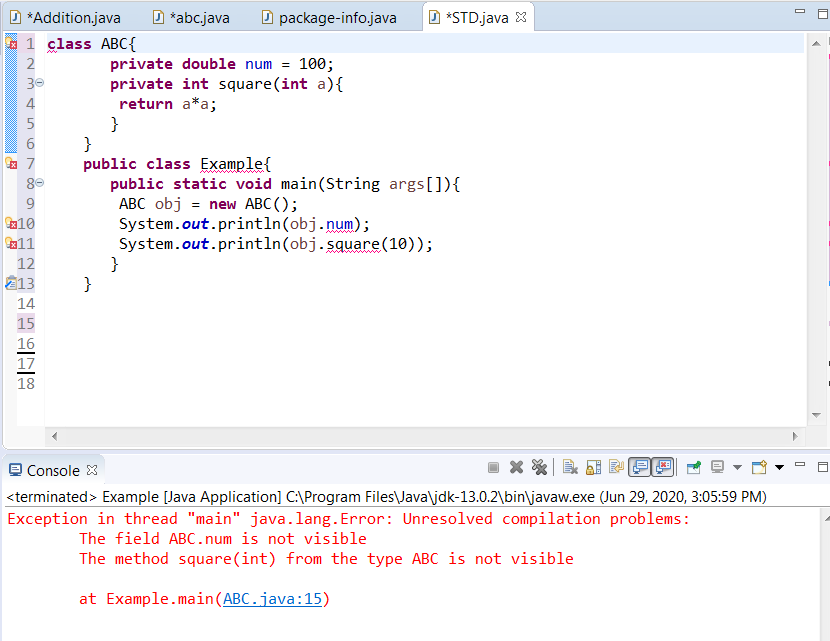
ANS (a): Access modifiers are used in java to control the accessibility to classes, interfaces, fields, constructors and methods. In other words, we can use access modifiers to protect data and behaviors from the outside world. At a high level, access modifiers facilitate the encapsulation feature in Object-Oriented Programming etc.

**Private Access Modifier:**

The scope of private modifier is limited the class only.

1. Private Data members and methods are only accessible within the class
2. Class and interface can’t be declared as private
3. If class and private constructor then we can’t create the object of that class from outside of the class.Private access modifier example in java

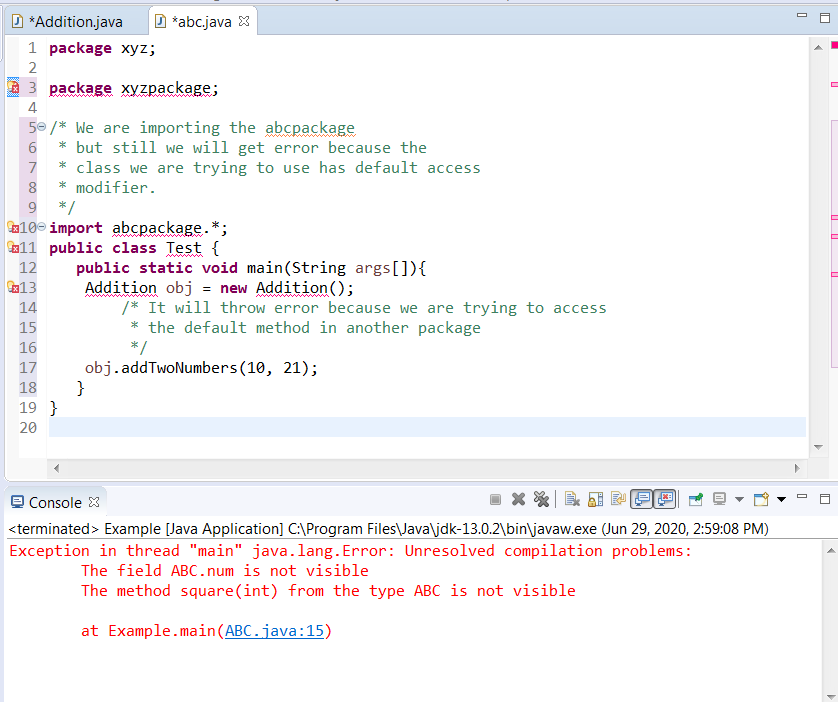
### This example throws compilation error because we are trying to access the private data member and method of class ABC in the class Example. The private data member and method are only accessible within the class.

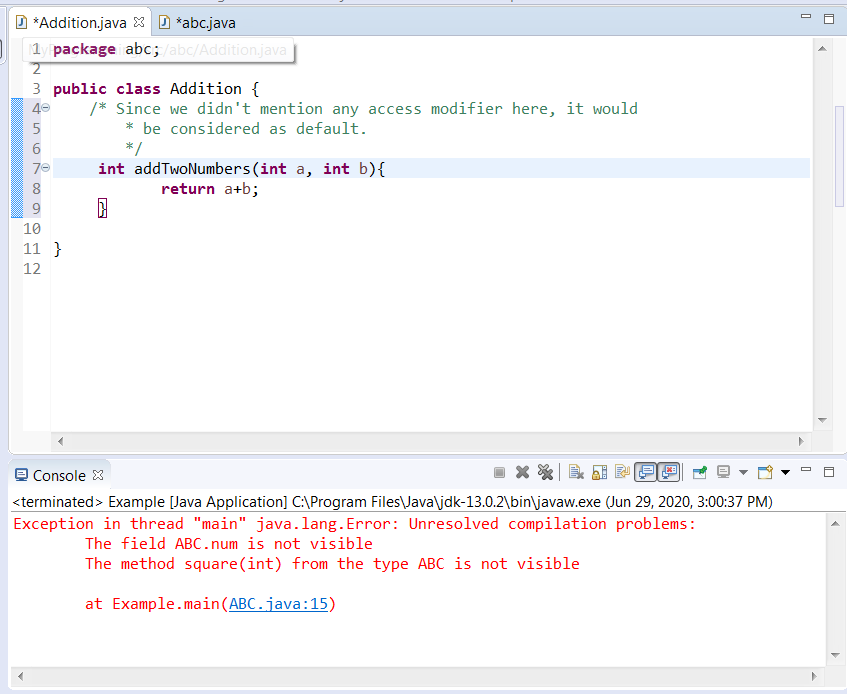


**Default Access Modifier:**

When we do not mention any access modifier, then,it is called default modifier.The scop of this modifier is limited to the package only.This means that if we have a class with the default access modifier in a package, only those classes that are in this package can access this class.No other class outside this package can access this class.Similarly if we have a default method or data member in a class, it would not be visible in the class of another package.Lets see an example to understand this:

**Default Access Modifier Example in Java:**

In this example we have two classes, Test class is trying to access the default method of Addition class, since class Test belongs to a different package, this program would throw compilation error, because the scope of default modifier is limited to the same package in which it is declared.



**END OF Q NO 1**

**Q NO 2: a. Explain in detail Public and Protected access modifiers?**

**b. Write a specific program of the above mentioned access modifiers in java.**

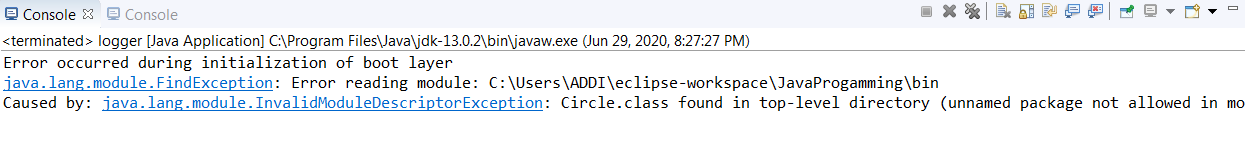
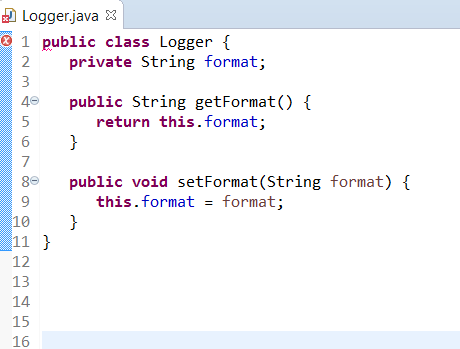
**ANS(a): Protected Access Modifier:**

The protected access modifier provides access within the package and outside the package through inheritance only i.e by creating child class (subclass or derived class). It cannot be accessed from outside the package if there is no inheritance.

**Protected Access Modifier Example:**  
  
In this example, the class Controller which is available in another package B can access the Multiply() method, which is declared as protected. This is possible since the class Controller extends class Multiplication. Subclasses of the Multiplication class present in any package can access the protected methods or data members.

### Public Access Modifier:

The public access modifier offers the lowest level of access restriction. Class or variable or method declared as public can be accessed everywhere. Once a variable or method is declared public, it can be accessed from anywhere in the code, i.e within the class, outside the class, within the package and outside the package.  
  
**Public Access Modifier Example:**  
  
The below example is equivalent to the one we have seen for private access modifier however here the method Multiply () has public modifier and class Controller can access it.



**Q NO3(a): What is inheritance and why it is used, discuss in detail ?**

**b. Write a program using Inheritance class on Animal in java.**

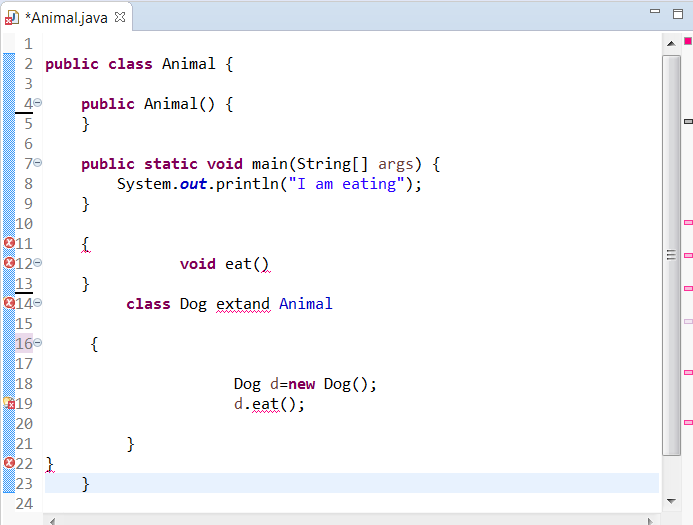
**ANS:(a): Inheritance:**

It is inheriting the properties of the parent class into child class

(OR) Inheritance is the procedure by which one object acquires all the properties and behaviors of a parent object.

**Using of inheritance in java:**

1. Inheritance is used in java to avoid from coding repeatation. It is used to reused the coding, if we write a program and in the program some coding are used repeatedly then we used the inheritance to avoid from repeating of the coding
2. Inheritance is used to achieve the polymorphism.Polymorphism is the process in which one task performed by deffrent methods.So by the help of iheritance we achieved the polymorphism
3. By inheritance we can achieve method overraiding.It means that one method is perform by another method.



(b):

**Q4. a. What is polymorphism and why it is used, discuss in detail ?**

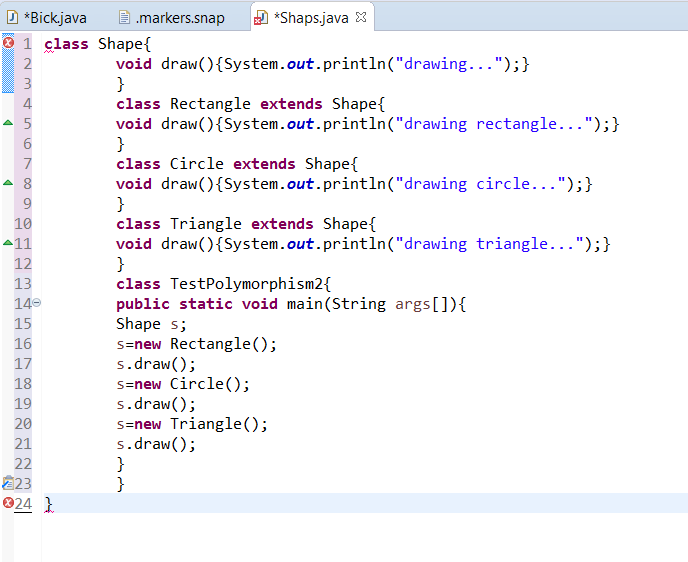
**b. Write a program using polymorphism in a class on Employee in java.**

**Ans**:Polymorphism a same task perform by deffrent methods is called polymorphism. Polymorphism is the combination of two woirds poly and morphism

‘‘Ploy’’ means many and ‘’ morphism ’’ means forms its means that different forms of a thing. For example: Sound is one entity but deffrent things perform these tasks by different ways.In java polymorphism is the method overloading and method overriding. In java if we performed a task by different methods is called polymorphism.It has two types

1.compile time poly morphism. Through compile time polymorphism we achieve method overloading.

2.Run time polymorphism:Through run time polymorphism we achieve method overriring.



**Q5. a. Why abstraction is used in OOP, discuss in detail ?**

**b. Write a program on abstraction in java.**

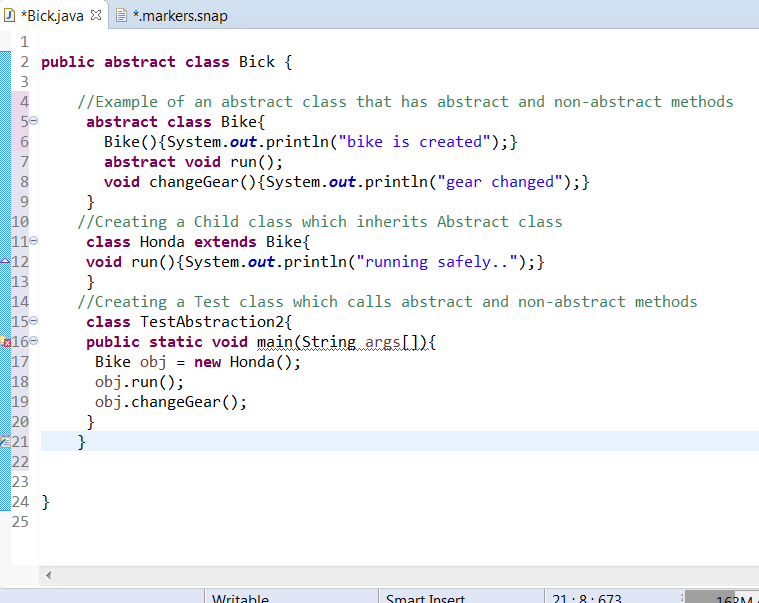
**ANS (a):** Abstraction is one of the [key concepts](https://stackify.com/oops-concepts-in-java/) of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details from the user. That enables the user to implement more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden complexity.

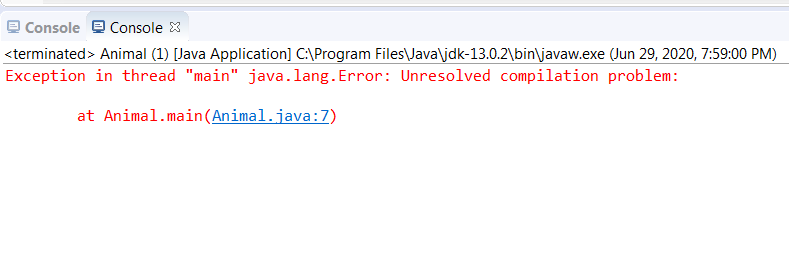
That’s a very generic concept that’s not limited to object-oriented programming. You can find it everywhere in the real world.

Abstraction in OOP

Objects in an OOP language provide an abstraction that hides the internal implementation details. Similar to the coffee machine in your kitchen, you just need to know which methods of the object are available to call and which input parameters are needed to trigger a specific operation. But you don’t need to understand how this method is implemented and which kinds of actions it has to perform to create the expected result.

Let’s implement the coffee machine example in Java. You do the same in any other object-oriented programming language. The syntax might be a little bit different, but the general concept is the same.

(b) 



**END OF Q NO 5**