**Name** **zarak saleem gul**

**Id** **16264**

**Department**  **SE**

**Section**   **(A)**

**Question 1:**

**a )**

**Access Modifiers:**

Access Modifiers is use for access control, when we use class in a program so inside the class members, variables and methods of the class can specify with the help of access modifiers. Access modifiers are also called visibility modifiers.

**Access Modifiers use on 2 levels:**

1. At the Top level

Like class Level access modifiers

1. At the Member Level

Like Member Level access modifiers

**Private Access Modifiers:**

In the above example we inherit class B with class A using keyword “extends” so in the class B we access all the variable of class A because here is no access modifiers use.

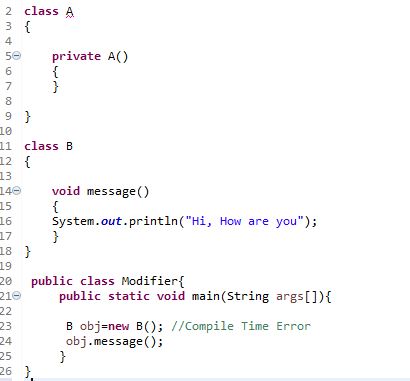
**Default Access Modifiers:**

The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

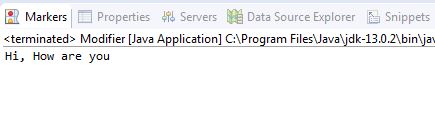
**b).**

**Program for Private Access modifiers:**

**Code:**

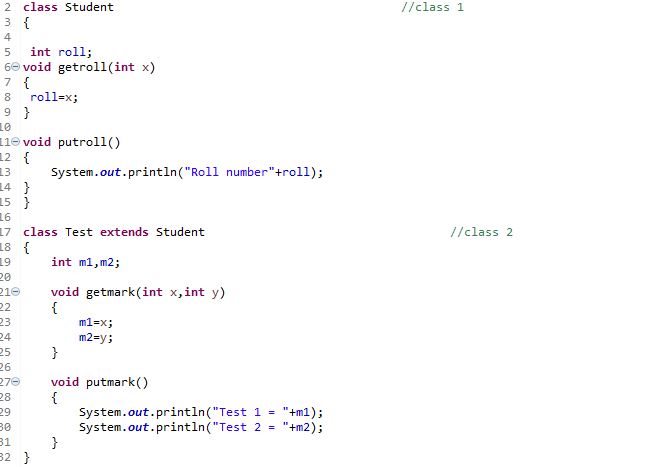


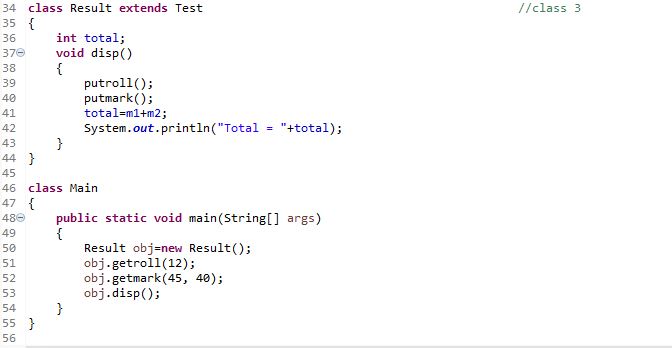
**Out put:**

****

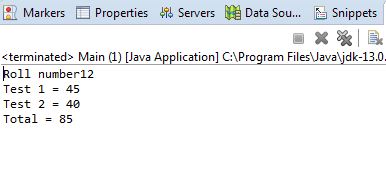
**Program for Default Access modifiers:**

**Code:**

****



**Output:**



**Question 2:**

**a)**

**Public access modifiers:**

The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package. Public access modifier access everywhere as you want.

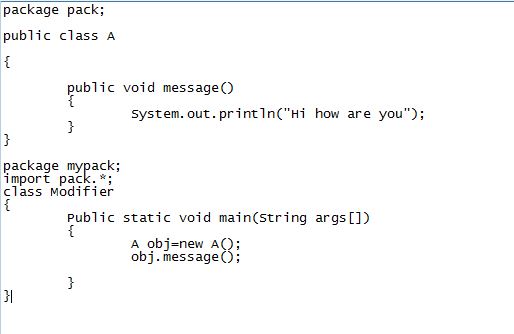
**Protected access modifiers:**

Variables, methods, and constructors, which are declared protected in a superclass can be accessed only by the subclasses in other package or any class within the package of the protected members' class. The protected access modifier cannot be applied to class and interfaces.

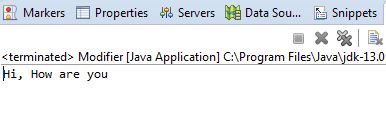
**b).**

**Program for public:**

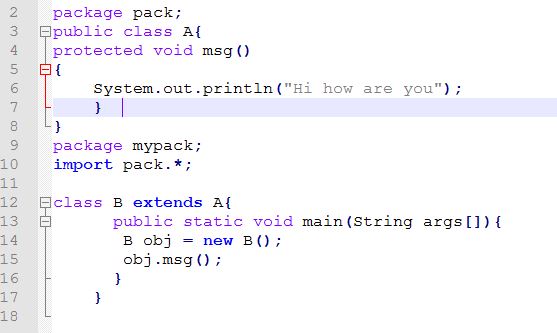
**Code:**

****

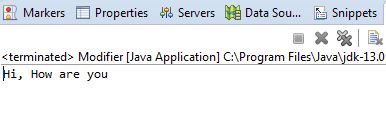
**Output:**

****

**Program for protected class:**

****

**Output:**

****

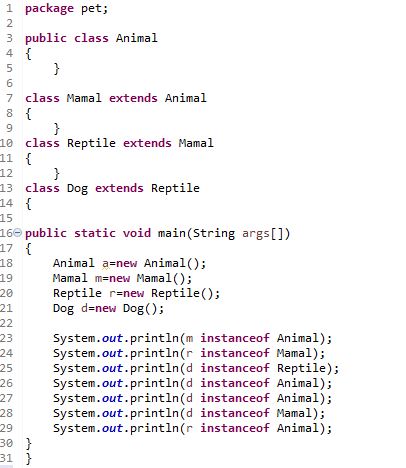
**Question 3:**

**a)**

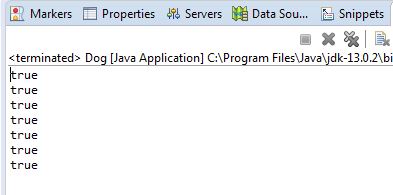
**Inheritance:**

Inheritance in Java Inheritance is an important pillar of OOP(Object Oriented Programming). It is the mechanism in java by which one class is allow to inherit the features (fields and methods) of another class. ... The subclass can add its own fields and methods in addition to the super class fields and methods.

**b)**



**Output:**

****

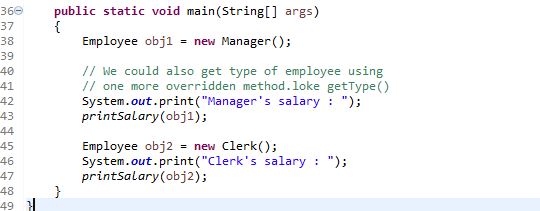
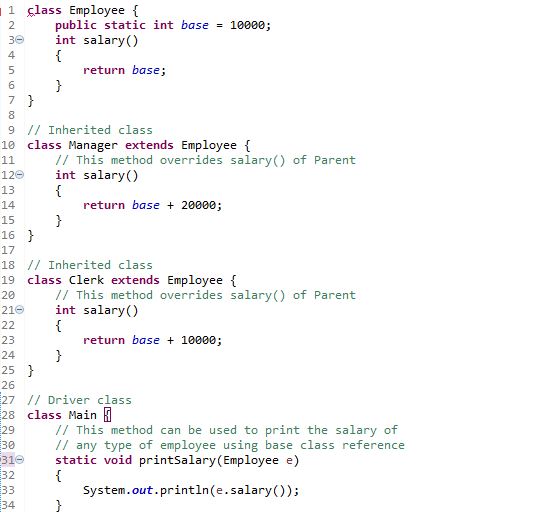
**Question 4:**

**a)**

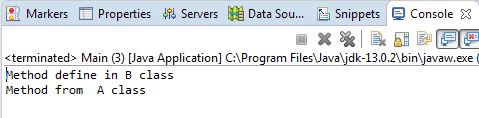
**Polymorphism:**

Polymorphism in Java is a concept by which we can perform a single action in different ways. We can perform polymorphism in java by method overloading and method overriding. If you overload a static method in Java, it is the example of compile time polymorphism. Here, we will focus on runtime polymorphism in java.

**b.**

****

**Output:**

****

**Question 5:**

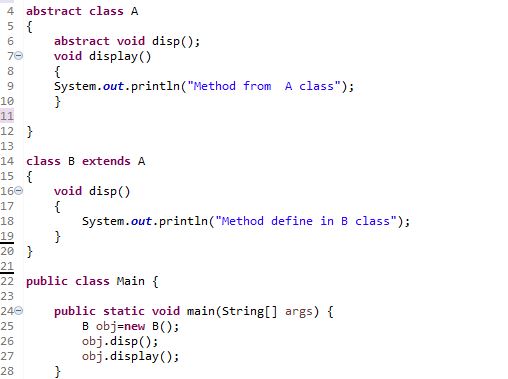
**a)**

**Abstraction**:

Abstraction is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message.

**b)**

**code:**



**Output:**

