**Question 1**

**Animals class:**

**public** **class** Animals {

**private** **static** Animals *animal*;

**private** Animals() {

}

**public** **static** Animals getInstance() {

**if**(**null**==*animal*)

{

*animal*=**new** Animals();

}

**return** *animal*;

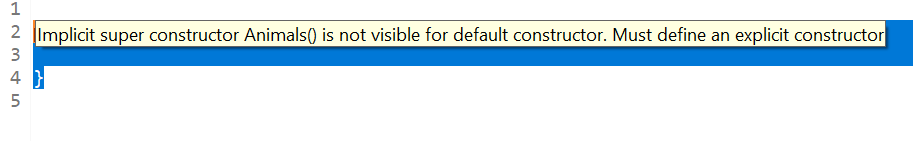
}

}

**Bird class:**

**public** **class** Birds **extends** Animals{

}



Shows singleton class cannot be inherited

**Question 2**

**Employee class:**

**public** **class** Employee {

String name;

**int** salary;

Employee(String name,**int** salary)

{

**this**.name=name;

**this**.salary=salary;

}

**public** **void** salary()

{

System.***out***.println("salary of Employee:"+salary);

}

}

**Manager class**:

**public** **class** Manager **extends** Employee{

**int** incentive;

Manager(String name, **int** salary,**int** incentive) {

**super**(name, salary);

**this**.incentive=incentive;

}

**public** **void** salary()

{

System.***out***.println("Salary of Manager:"+(salary+incentive));

}

}

**Labour class:**

**public** **class** Labour **extends** Employee {

**int** overTime;

Labour(String name, **int** salary, **int** ot) {

**super**(name, salary);

overTime=ot;

}

**public** **void** salary()

{

System.***out***.println("Salary of Labour:"+(salary+overTime));

}

}

**Main class:**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Manager m1=**new** Manager("Parth",30000,1000);

m1.salary();

Labour l1=**new** Labour("Aryan",20000,500);

l1.salary();

}

}

**Question 3**

**Account class:**

**public** **class** Accounts {

String name;

**int** cash;

Accounts(String name,**int** cash){

**this**.name=name;

**this**.cash=cash;

}

**int** balance() {

**return** cash;

}

}

**Savings class:**

**public** **class** savings **extends** Accounts {

**int** deposit;

savings(String n,**int** bal,**int** h){

**super**(n,bal);

deposit=h;

}

**int** balance() {

**return** (**super**.balance()+deposit);

}

}

**Current class:**

**public** **class** Current **extends** Accounts {

**int** cash;

Current(String n,**int** bal,**int** h){

**super**(n,bal);

cash=h;

}

**int** getSalary() {

**return** (**super**.balance()+cash);

}

}

**Main class:**

**public** **class** AccountMain {

**public** **static** **void** main(String[] args) {

savings s = **new** savings("Sneha",1000,5000);

System.***out***.println("Savings balance= "+s.balance());

Current c = **new** Current("Arya",0,0);

System.***out***.println("Current balance= "+c.balance());

}

}

**Question 4**

1.If any class has any of its abstract methods, then you must declare entire class abstract

**package** Assignment;

**public** **class** Animal {

String name;

**int** age;

**float** weight;

Animal(String n,**int** a,**float** w)

{

name=n;

age=a;

weight=w;

}

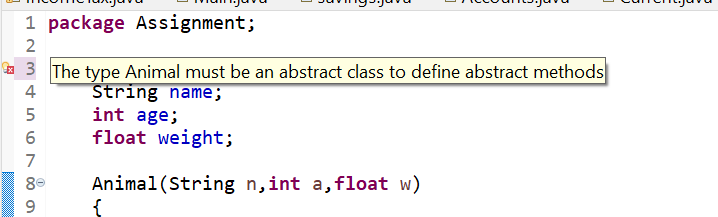
**abstract** **public** **void** eat();

**abstract** **public** **void** sleep();

**public** **static** **void** main(String[] args) {

}

}



2. Abstract class cannot be instantiated

**abstract** **public** **class** Animal {

**abstract** **public** **void** eat();

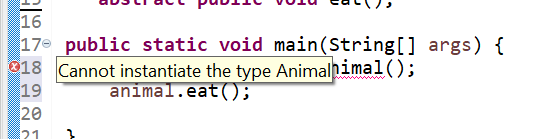
**public** **static** **void** main(String[] args) {

Animal animal= **new** Animal();

animal.eat();

}

}

****

3. When we extend abstract class, we must override all the methods of abstract class in sub class or declare subclasses as abstract

**public** **class** Lion **extends** Animal {

Lion(String n, **int** a, **float** w) {

**super**(n, a, w);

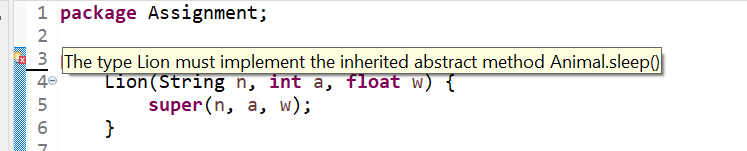
}

**public** **void** eat() {

System.***out***.println("eating..");

}

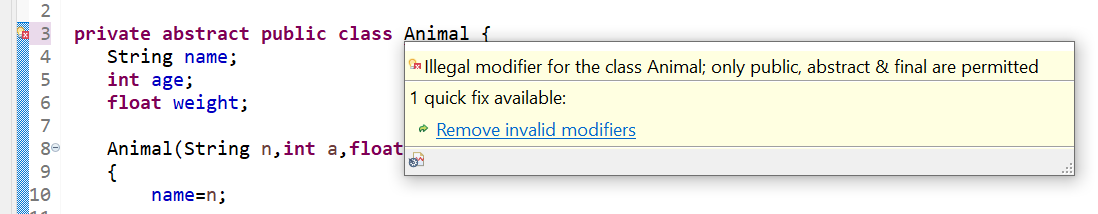
}



4. Abstract class cannot be private

**private** **abstract** **public** **class** Animal {

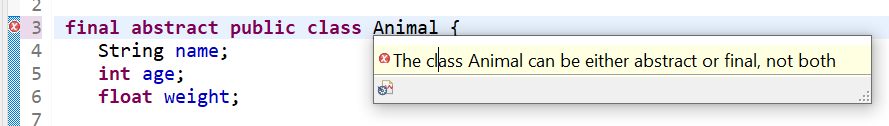
}



5. Abstract class cannot be final

**final** **abstract** **public** **class** Animal {

}



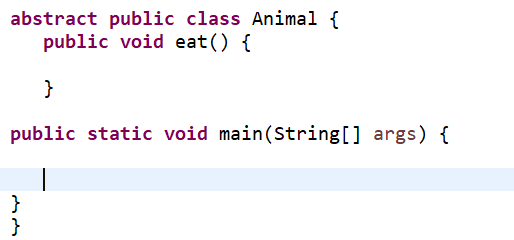
6.You can have an Abstract class without any abstract method

**abstract** **public** **class** Animal {

**public** **void** eat() {

}

}



**Question 5**

**Shape class:**

**package** Assignment;

**abstract** **public** **class** Shapes {

**abstract** **void** draw();

**public** **static** **void** main(String args[]) {

Line l=**new** Line();

l.draw();

Rectangle r= **new** Rectangle();

r.draw();

Cube c=**new** Cube();

c.draw();

}

}

**Line class:**

**public** **class** Line **extends** Shapes{

@Override

**void** draw() {

System.***out***.println("It has 2 points");

}

}

**Rectangle class:**

**public** **class** Rectangle **extends** Shapes {

@Override

**void** draw() {

System.***out***.println("It has 4 sides");

}

}

**Cube class:**

**public** **class** Cube **extends** Shapes{

@Override

**void** draw() {

System.***out***.println("It has 6 faces");

}

}