

**Iqra National University**

**Object Oriented Programming Final Paper**

**Nooriya Ilyas Khan | BS(SE) | 2<sup>nd</sup> Semester**

**ID: 15770**

**Course Instructor: Sir Ayub**

**Each question carries equal marks.  
Please answer briefly.**

**Question1.**

**a. Why access modifiers are used in java, explain in detail Private and Default access modifiers?**

**ANSWER:**

Access modifiers, as the name itself suggests, are keywords used in java programs in order to limit or restrict the accessibility of classes and it's members. In simpler words, the access to classes, constructors and methods, is controlled using the access modifiers.

There are 4 types of access modifiers. Public, Private, Default & Protected.

**Private:**

The Private access modifier is specified using the keyword private. The methods or data members declared as private are only accessible within the class in which they are declared in. Any other class of the same package will not have any access to them. Classes or interfaces cannot be declared as private.

**Default:**

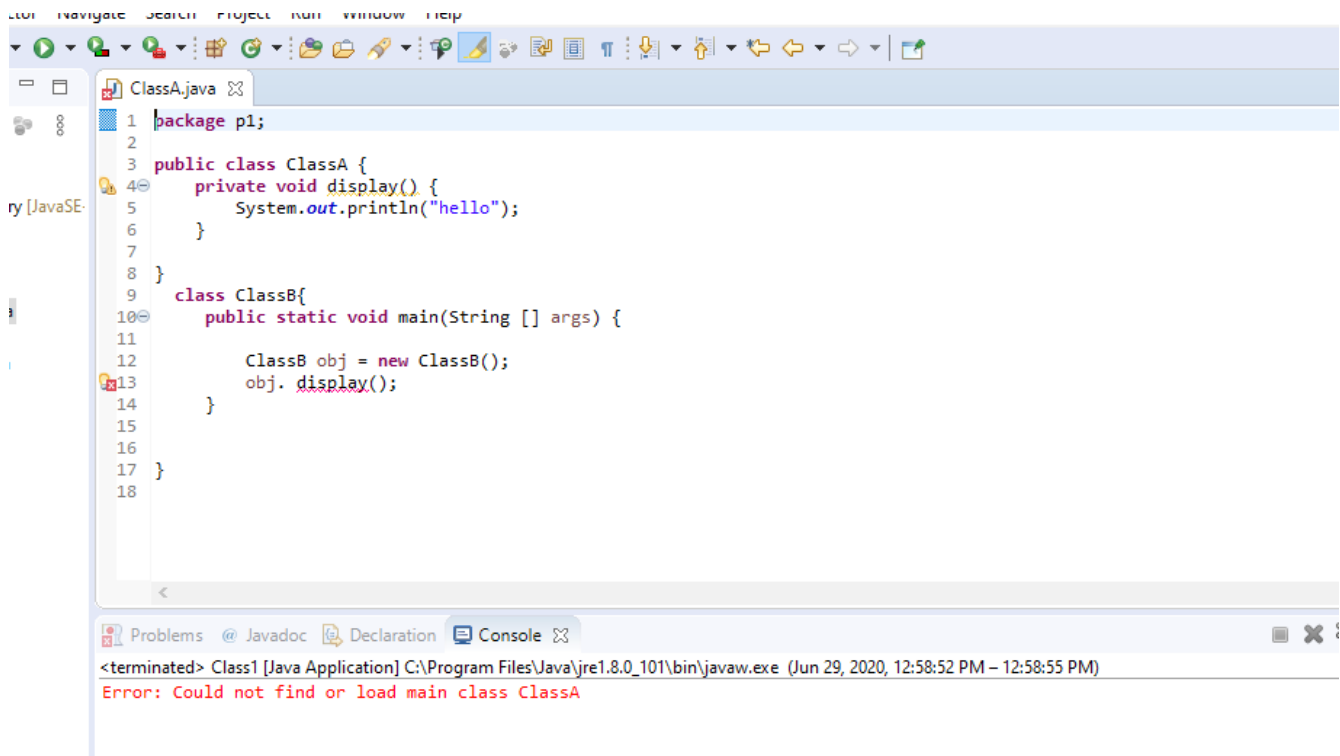
When no access modifier is specified for a class, method or data variable, it is said to have the default access modifier by default. Classes or methods which are not declared using any access modifiers are accessible only within the same package.

## Question1.

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

Answer:

Program for private access modifier:



```
1 package p1;
2
3 public class ClassA {
4     private void display() {
5         System.out.println("hello");
6     }
7 }
8
9 class ClassB{
10    public static void main(String [] args) {
11
12        ClassB obj = new ClassB();
13        obj.display();
14    }
15 }
16
17 }
18
```

Problems @ Javadoc Declaration Console

<terminated> Class1 [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 29, 2020, 12:58:52 PM – 12:58:55 PM)  
Error: Could not find or load main class ClassA

**Explanation:** The class “Class A” contains a private void method. This method can be only accessible inside the “Class A” that is, the class in which it is located. I tried to call this private method in Class B, but the compiler obviously could not access it and we got an error instead.

I wrote comments in the program to explain it better:

## Program for default access modifier:

```
Class1.java  Cake.java ✕
1 package p1;
2
3 class Cake // i did not declare this class with a specific access modifier which means it is default.
4           // because it is default, i cannot access this class or make an object of this class in the other file "Class1".
5 {
6     void message() { // i left this method unspecified too, so now it is also default.
7
8         System.out.println("I like fruit cake!");
9     }
10
11 }
```

Problems @ Javadoc Declaration Console ✕

```
<terminated> Class1 (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jun 29, 2020, 8:24:34 PM - 8:24:36 PM)
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
    The type Cake is not visible
    The type Cake is not visible
    The type Cake is not visible

    at p2.Class1.main(Class1.java:11)
```

```
Class1.java ✕  Cake.java
1 package p2;
2
3 import p1.*; // importing the package p1
4
5 class Class1 {
6
7     public static void main(String args[])
8     {
9
10
11     Cake obj = new Cake(); // this package does not know that a "Cake" class exists because of the default access modifier
12
13     obj.message(); // since the Cake class is invisible, it's method "message()" is also unaccessible
14
15     }
16
17
18 }
19
20
```

Problems @ Javadoc Declaration Console ✕

```
<terminated> Class1 (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jun 29, 2020, 8:18:46 PM - 8:18:48 PM)
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
    The type Cake is not visible
    The type Cake is not visible
    The type Cake is not visible

    at p2.Class1.main(Class1.java:11)
```

## **Question2.**

**a. Explain in detail Public and Protected access modifiers?**

**Answer:**

### **Public:**

The public access modifier is specified using the keyword public. The public access modifier has the widest scope among all the other access modifiers. Classes, methods declared as public are accessible from everywhere in the program. A class's public members are accessible wherever the program has a reference to an object of that class or one of its sub-classes.

### **Protected:**

The protected access modifier is specified by using the protected keyword. Using protected access modifier offers an intermediate level of access between public and private. The methods or data members declared as protected are accessible within same package or sub classes in different package. Classes cannot be declared protected.

## **Question2.**

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

**Answer:**

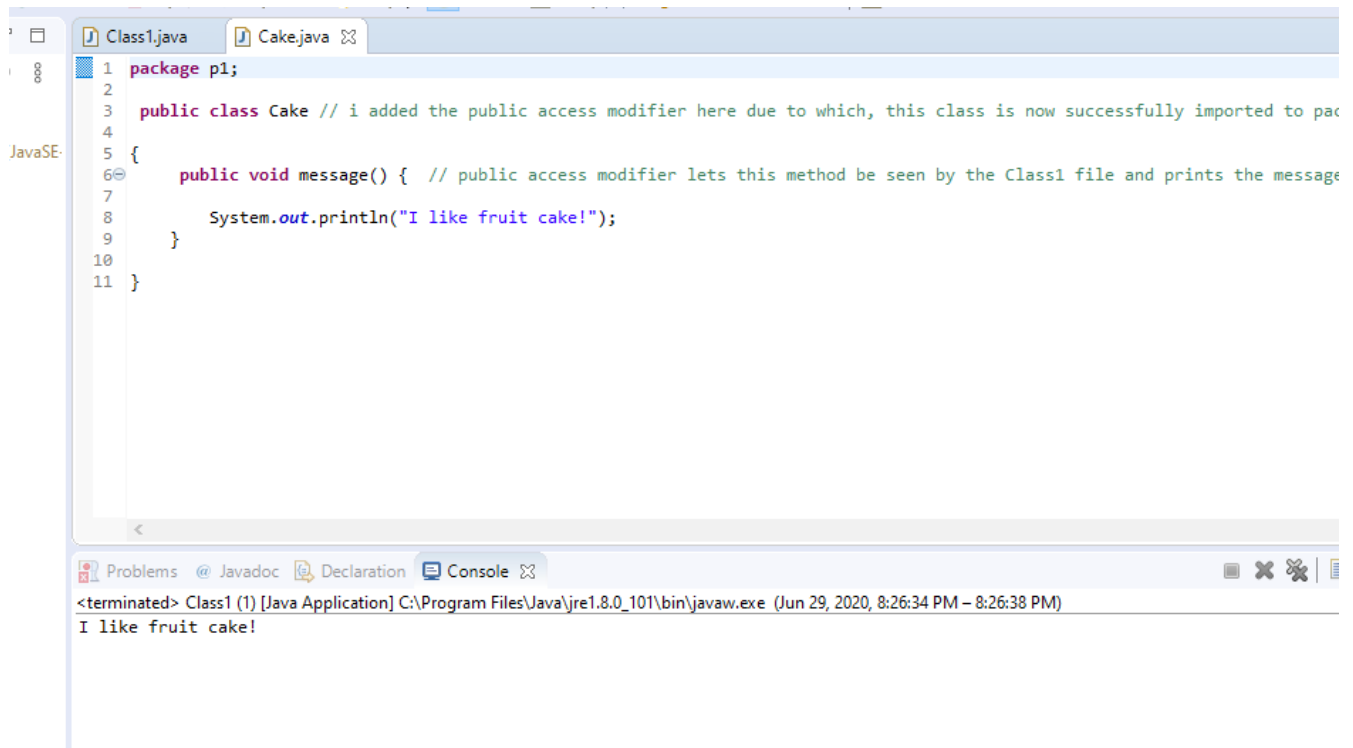
**Program of Public access modifier:**

## Explanation:

In the program below, I have created a class named `Cake` in package `p1`, and another class named `Class1` in package `p2`.

The `Cake` class contains a method that contains a message and I tried to access it through package `p2` by importing it.

The program ran just fine because the public declared methods and classes are accessible from everywhere.



```
1 package p1;
2
3 public class Cake // i added the public access modifier here due to which, this class is now successfully imported to pac
4
5 {
6     public void message() { // public access modifier lets this method be seen by the Class1 file and prints the message
7
8         System.out.println("I like fruit cake!");
9     }
10 }
11 }
```

Problems @ Javadoc Declaration Console

<terminated> Class1 (1) [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 29, 2020, 8:26:34 PM – 8:26:38 PM)

I like fruit cake!

```
Class1.java Cake.java
1 package p2;
2
3 import p1.*;
4
5 class Class1 {
6
7     public static void main(String args[])
8     {
9
10        Cake obj = new Cake();
11
12        obj.message();
13
14    }
15
16
17
18 }
19
20
```

Problems @ Javadoc Declaration Console

<terminated> Class1 (1) [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 29, 2020, 8:26:34 PM – 8:26:38 PM)  
I like fruit cake!

### Program for protected access modifier:

Two classes, the parent class(Apples) and child class(Pie) are shown below. The parent class has a protected method which is accessible by it's sub-class(Pie) despite being in another package.

```
Apples.java Pie.java
1 package p2;
2
3 public class Apples {
4
5     protected void display() {
6         System.out.println("Green apples!");
7     }
8 }
9
10
```

Problems @ Javadoc Declaration Console

<terminated> Pie [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 29, 2020, 11:31:26 PM – 11:31:29 PM)  
Green apples

```
Apples.java Pie.java
1 package p1;
2
3 import p2.Apples;
4
5 class Pie extends Apples {
6     public static void main( String[]args) {
7         Pie obj = new Pie();
8         obj.display();
9     }
10 }
```

Problems @ Javadoc Declaration Console

<terminated> Pie [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 29, 2020, 11:32:12 PM – 11:32:14 PM)

Green apples!

### Question3.

a. What is inheritance and why it is used, discuss in detail?

**Answer:**

Inheritance is an operation that allows a class to inherit properties of another class. When a subclass (child class) extends a super-class (parent class) it inherits all the non-private members including fields and methods. Inheritance is an effective method to share code between classes that have some traits in common, yet allowing the classes to have some parts that are different.

**Syntax:**

```
class childclass extends parentclass
{
    // methods() & fields
}
```

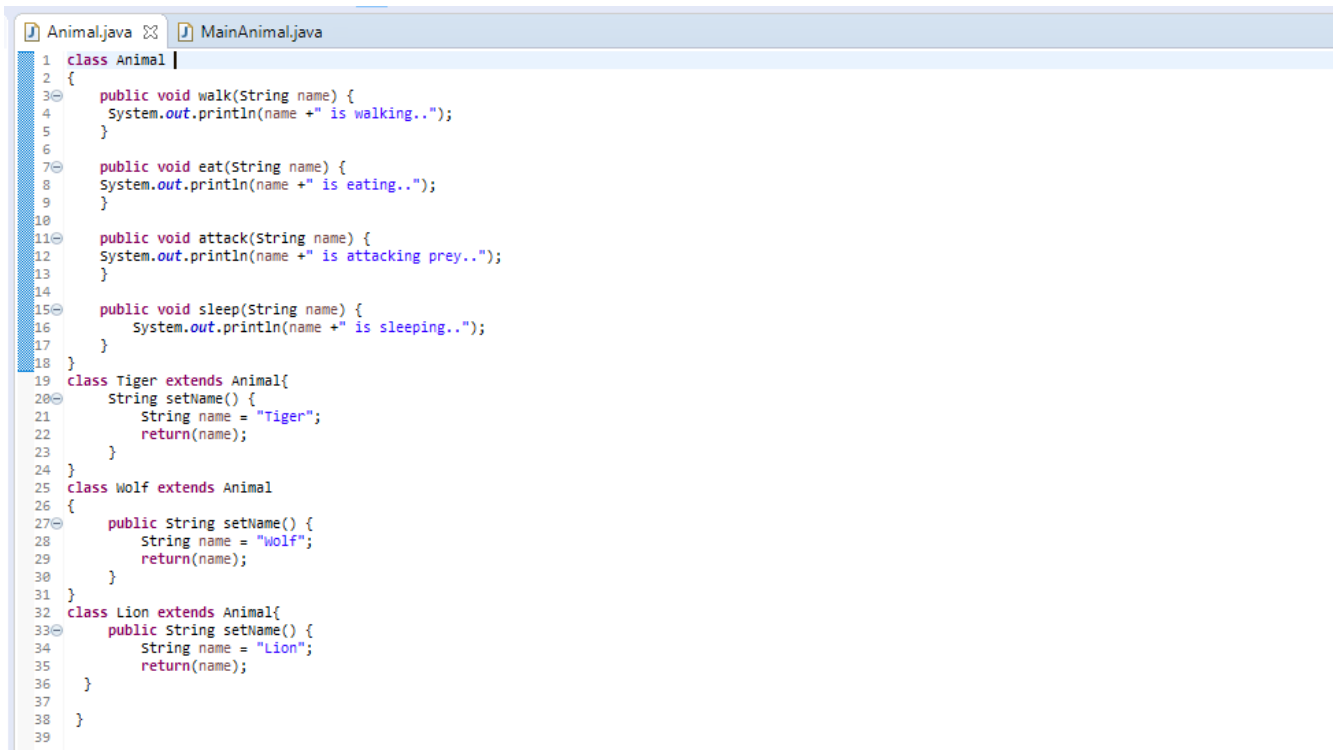


### Question3.

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

#### Program using inheritance:

In the program below I created two separate files, one contains the parent classes, their child classes, and methods. While the MainAnimal class contains the main method that will create objects and call the methods and execute the program.



```
1 class Animal {
2 {
3     public void walk(String name) {
4         System.out.println(name + " is walking..");
5     }
6
7     public void eat(String name) {
8         System.out.println(name + " is eating..");
9     }
10
11    public void attack(String name) {
12        System.out.println(name + " is attacking prey..");
13    }
14
15    public void sleep(String name) {
16        System.out.println(name + " is sleeping..");
17    }
18 }
19 class Tiger extends Animal{
20     String setName() {
21         String name = "Tiger";
22         return(name);
23     }
24 }
25 class Wolf extends Animal
26 {
27     public String setName() {
28         String name = "Wolf";
29         return(name);
30     }
31 }
32 class Lion extends Animal{
33     public String setName() {
34         String name = "Lion";
35         return(name);
36     }
37 }
38 }
39
40
```

Here we see the output. The child classes (Tiger,Wolf,Lion) inherited all the methods present in the parent class (Animal).

```
1
2 class MainAnimal {
3
4 public static void main( String[]args) {
5     Tiger t1 = new Tiger();
6     Wolf w2 = new Wolf();
7     Lion L3 = new Lion();
8
9     t1.walk(t1.setName());
10    t1.eat(t1.setName());
11
12    w2.attack(w2.setName());
13    w2.sleep(w2.setName());
14
15    L3.walk(L3.setName());
16    L3.attack(L3.setName());
17
18 }
19
20
```

Problems @ Javadoc Declaration Console

<terminated> Pie [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 30, 2020, 1:48:29 AM – 1:48:31 AM)

```
Tiger is walking..
Tiger is eating..
Wolf is attacking prey..
Wolf is sleeping..
Lion is walking..
Lion is attacking prey..
```

#### Question4.

a. What is Polymorphism and why it is used, discuss in detail ?

Answer:

Polymorphism is considered one of the most important features of object oriented programming. The term Polymorphism means having many forms. Polymorphism in java does exactly what it's name suggests; it allows us to perform a single action in multiple and different ways. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object. Inheritance allows code to be re-used and the polymorphism is, the occurrence of one function with different forms.

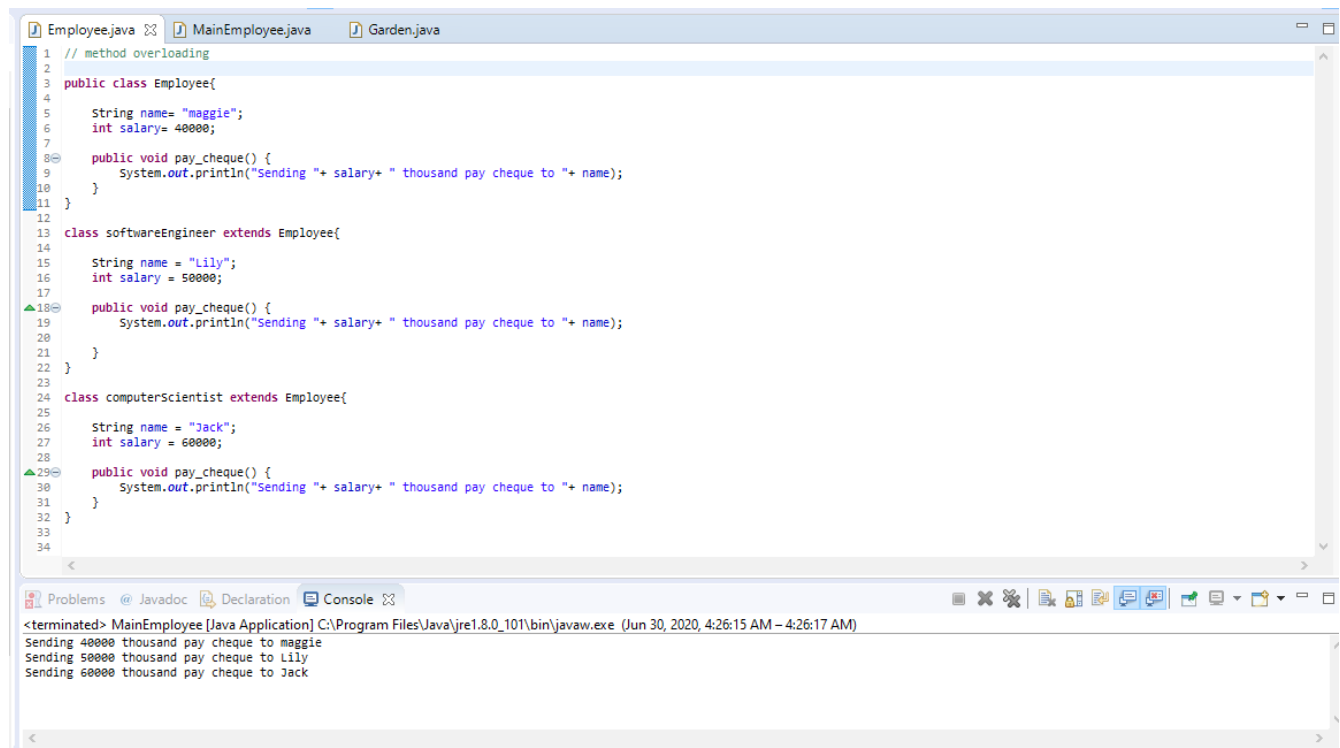
## Question4.

b. Write a program using Polymorphism in a class on Employee in java.

Answer:

I created two examples of polymorphism, one is method overloading, in which a single method is used over and over again but it has different value and message each time it is called.

**Example of Method overloading:**



```
Employee.java MainEmployee.java Garden.java
1 // method overloading
2
3 public class Employee{
4
5     String name= "maggie";
6     int salary= 40000;
7
8     public void pay_cheque() {
9         System.out.println("Sending "+ salary+ " thousand pay cheque to "+ name);
10    }
11 }
12
13 class softwareEngineer extends Employee{
14
15     String name = "Lily";
16     int salary = 50000;
17
18     public void pay_cheque() {
19         System.out.println("Sending "+ salary+ " thousand pay cheque to "+ name);
20     }
21 }
22
23
24 class computerScientist extends Employee{
25
26     String name = "Jack";
27     int salary = 60000;
28
29     public void pay_cheque() {
30         System.out.println("Sending "+ salary+ " thousand pay cheque to "+ name);
31     }
32 }
33
34
```

Problems @ Javadoc Declaration Console

```
<terminated> MainEmployee [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jun 30, 2020, 4:26:15 AM - 4:26:17 AM)
Sending 40000 thousand pay cheque to maggie
Sending 50000 thousand pay cheque to Lily
Sending 60000 thousand pay cheque to Jack
```

**The main method of method overloading example:**

```
Employee.java MainEmployee.java Garden.java
1 public class MainEmployee {
2
3
4     public static void main(String[]args) {
5         Employee obj1 = new Employee();
6         Employee obj2 = new softwareEngineer();
7         Employee obj3 = new computerScientist();
8
9         obj1.pay_cheque();
10        obj2.pay_cheque();
11        obj3.pay_cheque();
12
13    }
14 }
15
```

Problems @ Javadoc Declaration Console

<terminated> MainEmployee [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 30, 2020, 4:26:15 AM – 4:26:17 AM)  
Sending 40000 thousand pay cheque to maggie  
Sending 50000 thousand pay cheque to Lily  
Sending 60000 thousand pay cheque to Jack

This is the second example, the operator overloading, in which a same method has different parameters.

### Example of Operator overloading:

```
Employee.java MainEmployee.java Garden.java
1 // operator overloading
2
3 public class Garden {
4
5     void Plants (int a, int b) {
6         int sum = a + b;
7         System.out.println("Total number of plants in the garden is " + sum);
8
9     }
10
11
12
13     void Plants (String a, String b, String c) {
14         System.out.println("The evergreen plants of the garden are "+a+ ", "+b+ " and "+c);
15     }
16     public static void main(String[]args) {
17         Garden obj = new Garden();
18         obj.Plants(5, 10);
19         obj.Plants("poppy ", " snake plant ", "lily");
20
21     }
22 }
23
24 }
25
26
27
```

Problems @ Javadoc Declaration Console

<terminated> Garden [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 30, 2020, 4:24:58 AM – 4:24:59 AM)  
Total number of plants in the garden is 15  
The evergreen plants of the garden are poppy , snake plant and lily

## **Question5.**

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

**Answer:**

Abstraction is a process of hiding the implementation details and showing only the functionality to the user. In simpler words, abstraction is used in OOP to show only the important and relevant stuff to the user while hiding the internal details that are of no use to the user.

In Java, abstraction is carried out through Abstract classes, Abstract methods, and interfaces. Abstraction helps in reducing complications and hardwork in programming.

**Abstract Class:**

A class declared using the keyword abstract is called as an abstract class. It can have abstract methods as well as concrete methods. A normal class cannot have abstract methods.

**Abstract Method:**

An Abstract Method is a method without a body. It must be declared in an abstract class. The abstract method will never be final because the abstract class must implement all the abstract methods.

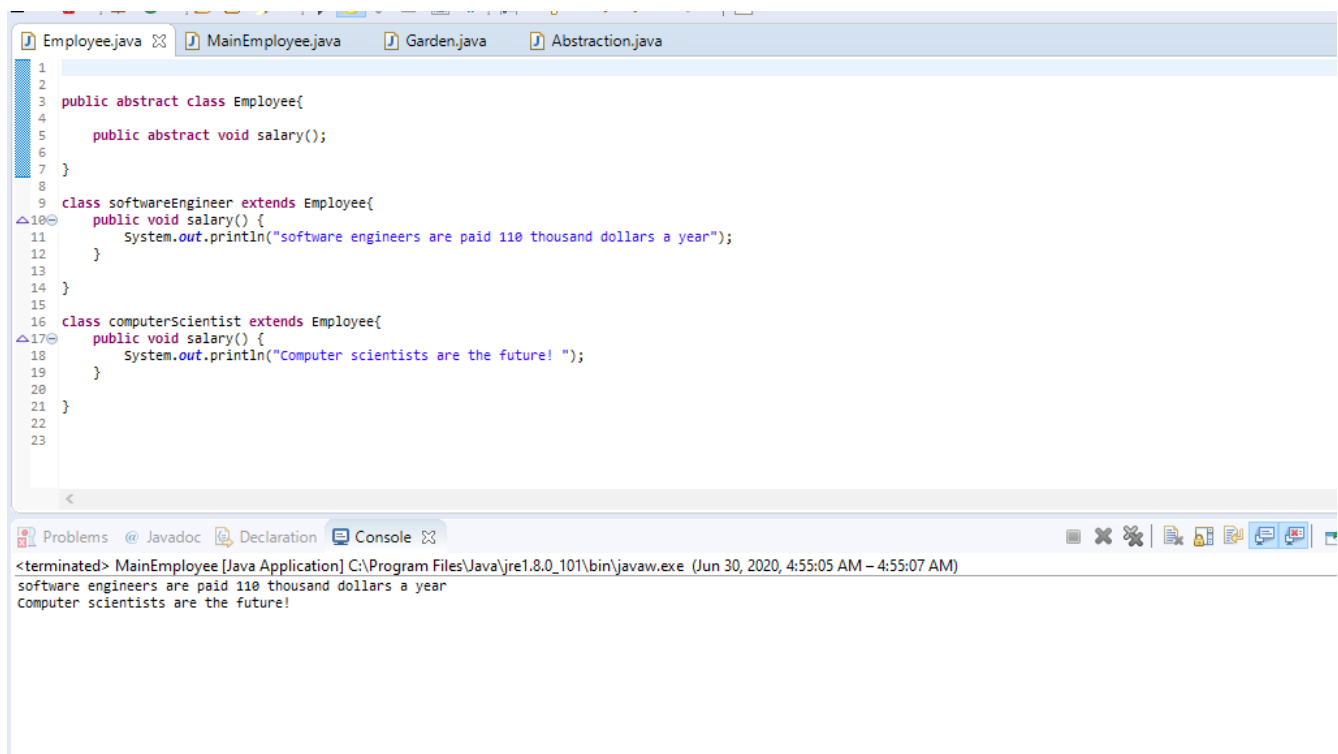
- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- It cannot be instantiated.
- It can have constructors and static methods too.
- It can have final methods which will force the subclass not to change the body of the method.

## Question5.

b. Write a program on abstraction in java.

Answer:

I modified one of the programs I wrote for the polymorphism example by creating abstract class and method by writing the abstract keyword with the class Employee, and I created an empty abstract method. I wrote the code in the file named Employee.java and I wrote the main method for this code in another file named MainEmployee.java.



```
Employee.java MainEmployee.java Garden.java Abstraction.java
1
2
3 public abstract class Employee{
4
5     public abstract void salary();
6
7 }
8
9 class softwareEngineer extends Employee{
10     public void salary() {
11         System.out.println("software engineers are paid 110 thousand dollars a year");
12     }
13 }
14
15 class computerScientist extends Employee{
16     public void salary() {
17         System.out.println("Computer scientists are the future! ");
18     }
19 }
20
21 }
22
23
<terminated> MainEmployee [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jun 30, 2020, 4:55:05 AM - 4:55:07 AM)
software engineers are paid 110 thousand dollars a year
computer scientists are the future!
```

The main method:

```
Employee.java MainEmployee.java Garden.java Abstraction.java
1
2 public class MainEmployee {
3
4     public static void main(String[] args) {
5         Employee obj2 = new softwareEngineer();
6         Employee obj3 = new computerScientist();
7
8         obj2.salary();
9         obj3.salary();
10
11     }
12 }
13
14
```

Problems @ Javadoc Declaration Console

<terminated> MainEmployee [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jun 30, 2020, 4:55:05 AM - 4:55:07 AM)  
software engineers are paid 110 thousand dollars a year  
Computer scientists are the future!