

Assignment No. 03 (Unit 3)

Q1) Write a Java program to calculate the area of rectangle, circle and triangle using the concept of abstract class.

→

Code:

```
package com.company;
```

```
import java.util.*; // import
```

```
abstract class Area {
```

```
final float PI = 3.14f;
```

```
float half final float half = 0.5f;
```

```
abstract float area (float a, float b);
```

{}

```
class Rectangle extends Area {
```

```
float area (float length, float breadth) {
```

```
return length * breadth;
```

{}

{}

```
class Circle extends Area {
```

```
float area (float radius, float notRequired) {
```

```
return PI * radius * radius;
```

{}

{}

```
class Triangle extends Area {
```

```
float area (float height, float base) {
```

```
return half
```

return $\frac{1}{2} \times \text{length} \times \text{breadth} \times \text{height}$; (2)

3. ~~square~~ \rightarrow ~~rectangle~~ \rightarrow ~~triangle~~ \rightarrow ~~circle~~

3

new ~~Circle~~ \rightarrow ~~square~~ \rightarrow ~~rectangle~~ \rightarrow ~~triangle~~ \rightarrow ~~circle~~

refinement class $\Phi_1 \Sigma$ how abstraction work

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

~~Rectangle r1 = new Rectangle();~~

~~System.out.println("Area of Rectangle : " +~~

~~r1.area(10, 20));~~

~~Circle c1 = new Circle(10, 0);~~

~~System.out.println("Area of Circle : " +~~

~~c1.area(10, 0));~~

~~Triangle t1 = new Triangle();~~

~~System.out.println("Area of Triangle : " +~~

~~t1.area(10, 20));~~

~~}~~

~~3. ~~square~~ \rightarrow ~~rectangle~~ \rightarrow ~~triangle~~ \rightarrow ~~circle~~~~

Output : [CMD]

javac	Φ_1 .java
-------	----------------

java	Φ_1
------	----------

Area of Rectangle : 200.0	
---------------------------	--

Area of Circle : 314.0	
------------------------	--

Area of Triangle : 100.0	
--------------------------	--

Q2) Define an interface. Write a program which illustrates the design and implementation of an interface:

→

Interface: "Interfaces are basically a class which can have methods and variables with a major difference being that its interfaces define only abstract methods and final fields (data members)."

+ : ~~signature is empty") nothing, due, methods~~

Code: ~~(((class, or, interface, etc))~~

(((interface Employee {

int static final int diwaliBonus = 20000;
int float compute(float salary);

((3) point class = to implement

+ : Employee ? class AProgrammer implements Employee {

((class, or, interface, etc))

int float compute(float salary) {

return salary + diwaliBonus;

3

3

: output

public class

class Q2 { : ((empty, to, run))

public static void main(String[] args) {

Programmer p = new Programmer();

System.out.println("Salary : " + p.compute(50000));

3

Output :

CMO

javac Q2.java

java Q2

Salary : 70000

Q.3) a) What is a package? Explain member access privileges.

Package - Packages are Java language's way of grouping a variety of classes and/or interfaces together.

Java packages act as "containers" for classes.

Java packages are further classified into two types :

1) Java API package

2) User defined packages.

Member access privileges:

1) public

2) protected

3) default (friendly)

4) private

5) protected

2) Public :

"Sharing" here "means" Any variable or method is visible to the entire class in which it is defined.

Public :

- A variable or method declared public has the widest possible visibility and is accessible everywhere.
- Implemented by just adding public keyword before e.g. public int number.

Protected :

- Visibility level is between public and protected.
- The protected modifier makes the fields visible to all classes and sub classes in the same package, but also to subclasses in other packages.
- Non-subclasses in other packages cannot access the protected members.

Default :

- Also known as "friendly" level of access.
- Makes fields visible in all classes of the same package.
- The fields are not visible in other packages.

Private protected :

• Declaration:

private protected int age;

- Visibility is between "protected" and "private".
- The fields are visible to all subclasses regardless of the package.

- These fields are not accessible by other classes.

Private:

- Fields are accessible only by the class itself.
- These fields cannot be inherited by subclasses and hence, are not accessible in subclasses.

Access	Public	Protected	Friendly (default)	Private	Private
Location				Protected	
Same class	Yes	Yes	Yes	Yes	Yes
Subclass in same package	Yes	Yes	Yes	Yes	No
Subclass in other packages	Yes	Yes	No	Yes	No
Other classes in same package	Yes	Yes	Yes	No	No
Non subclass in other packages	Yes	No	No	No	No

3.a.1. Visibility of fields.

b) Write a sample program to illustrate packages.

→

Code:

~~class Area {~~
~~float area() {~~
~~return pi * r * r;~~
~~}~~
~~}~~

~~class Circle {~~
~~float diameter() {~~
~~return 2 * pi * radius;~~
~~}~~
~~}~~

~~public class Area {~~
~~abstract float area();~~
~~}~~

~~public class Circle {~~
~~float diameter();~~
~~float radius();~~
~~float area();~~
~~}~~

Code:

package mathematics;

public class circle {

 float diameter();

 float radius();

 float area();

circle (float radius) {

 this.radius = radius;

 diameter = 2 * radius;

void CalcArea() {

 System.out.println("Area : " + (3.14 *
 radius * radius));

3

3

Code 2:

```
import mathematics.Circle;
class Q3 {
    public static void main(String[] args) {
        Circle c1 = new Circle(10);
        c1.CalcArea();
    }
}
```

Output:

```
C:\Users\Aman bhai\studeying>javac Q3.java
C:\Users\Aman bhai\studeying>java Q3; area = 314.0
C:\Users\Aman bhai\studeying>
```

Q4) Explain about static import with an example.

→

- The static import feature eliminates the need of qualifying a static member with the class name.
- We can use the static import statement to import static members from classes and use them without qualifying the class name.

Syntax:

import static package-name :: subpackage-name . class-name
static-member-name;

Example:

import static java.lang.Math.*;

OR import static java.lang.Math.PI;

Code:

import static java.lang.Math.*;

public class Q4 {

 class Q4 {

 public static void Main(String[] args) {

 System.out.print("Enter radius : ");

 Scanner sc = new Scanner(System.in);

 int radius = sc.nextInt();

 float area = PI * radius * radius;

 System.out.print("Area : " + area);

}

3

Output: (CMD)

javac Q4.java	to compile program into .class file
---------------	-------------------------------------

java Q4	to run the program
---------	--------------------

Enter radius :	10
----------------	----

Area :	314.0
--------	-------

Q5) Write a corrected code/program for the details/information given below.

Solution:

Code:

package p1; // different file

public class Teacher;

{.....}

public class Student

{.....}

package p2; // different file

public class Courses.

{.....}

public class Student

{.....}

import p1.* // different file

import p2.*

p1.Student student1;

OR

p2.Student student2;