Assginment3	
Assignment name:	Array of Objects and Packages.

Array of Objects: - An array that conations **class type elements** are known as an **array of objects**. It stores the

reference variable of the object. Before creating an array of objects, we must create an instance of the class by using

the new keyword. Suppose, we have created a class named Student. We want to keep marks of 20 students. In this

case, we will not create 20 separate objects of Student class. Instead of this, we will create an array of objects, as

follows:

```
Student s[] = new Student[20];

public class Student

{

int marks;

}

public class ArrayOfObjects

{

public static void main(String args[])

{

Student std[] = new Student[3];// array of reference variables of Student

std[0] = new Student(); // convert each reference variable into Student object std[1] = new Student();

std[2] = new Student();

std[0].marks = 40; // assign marks to each Student element

std[1].marks = 50;

std[2].marks = 60;

System.out.println("\n students average marks:"

+(std[0].marks+std[1].marks+std[2].marks)/3);

}

}
```

Constructors: Constructor is a block of codes similar to the method. It is called when an instance of the class is

created. At the time of calling constructor, memory for the object is allocated in the memory. It is a special type of

method which is used to initialize the object. Every time an object is created using the new() keyword, at least one

constructor is called. It calls a default constructor if there is no constructor available in the class. In such case, Java

compiler provides a default constructor by default.

Java Default Constructor

A constructor is called "Default Constructor" when it doesn't have any parameter.

Syntax of default constructor:

```
[accessspefier] <class_name>()
{
}
public class Student
{
private int id;
private String name; public Student() //Default constructor
{
id=0;
name="";
}
public void display()
{
System.out.println(id+" "+name);
}
public static void main(String args[])
{
Student s1=new Student();
s1.display();
}
```

}

Java Parameterized Constructor

A constructor which has a specific number of parameters is called a parameterized constructor. The parameterized

constructor is used to provide different values to distinct objects. However, you can provide the same values also.

```
public class Student

{

private int id;

private String name;

public Student(int i, String n) //creating a parameterized constructor

{

id = i;

name = n;

}

public void display () //method to display the values

{

System.out.println(id+" "+name);

}

public static void main(String args[])

{

//creating objects and passing values

Student s1 = new Student(111,"Karan");

Student s2 = new Student(222,"Aryan");

//calling method to display the values of object

s1.display(); s2.display();

}

}
```

this keyword: This keyword can be used to refer to the current class instance variable. If there is ambiguity between

the instance variables and parameters, this keyword resolves the problem of ambiguity.

```
import java.io.*;
```

```
public class Student
{
private int rollno;
private String name;
private float fee;
public Student(int rollno,String name,float fee)
{
this.rollno=rollno;
this.name=name;
this.fee=fee;
}
public void display()
{
System.out.println(rollno+" "+name+" "+fee);
}
public static void main(String args[])
{
Student s1=new Student(111,"ankit",5000);
s1.display();
}
}
```

Package

A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantage of Java Package

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

2) Java package provides access protection.

3) Java package removes naming collision.

Simple example of java package

The package keyword is used to create a package in java.

```
//save as Simple.java

package mypack;

public class Simple

{

public static void main(String args[])

{

System.out.println("Welcome to package");

}

}
```

How to compile java package

If you are not using any IDE, you need to follow the **syntax** given below:

javac -d directory javafilename

For **example**

javac -d . Simple.java

The -d switch specifies the destination where to put the generated class file. You can use any directory name

like /home (in case of Linux), d:/abc (in case of windows) etc. If you want to keep the package within the

same directory, you can use . (dot).

How to run java package program -

You need to use fully qualified name e.g. mypack.Simple etc to run the class.

Lab Assignment

SET A

1. Write a Java program to print the sum of elements of the array. Also display array elements in ascending order.

2. Write a Java program create class as MyDate with dd,mm,yy as data members. Write default and parameterized constructor. Display the date in dd-mm-yy format.(Use this keyword)

SET B

1. Define a class MyNumber having one private integer data member. Write a default constructor initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative, isPositive, isOdd, iseven. Use command line argument to pass a value to the object and perform the above tests.

2. Write a java program which define class Employee with data member as name and salary. Program store the information of 5 Employees. (Use array of object)

3. Write a java program to create class Account (accno, accname, balance). Create an array of “n” Account objects. Define static method “sortAccount” which sorts the array on the basis of balance. Display account details in sorted order.

Signature of the instructor: _____ Date: _____

Assignment Evaluation

0: Not Done [ ] 1: Incomplete [ ] 2: Late Complete [ ]

3: Needs Improvement [ ] 4: Complete [ ] 5: Well done [ ]