Assignment- 4

1] Explain use of interface with suitable example.

-> Interface is can have methods and variables, but the methods declared in an interface are by default abstract

-> we use interface because,

- It is used to achieve total abstraction.
- Since java does not support multiple inheritance in case of class, but by using inteface it can achieve multiple inheritance.
- It is also used to achieve loose coupling.
- Interface are used to implement abstraction. So the question arises why use interfaces when we have abstract class? The reason is, abstract class may contain non-final variables, wheres variables in interface are final, public and static.

Example.

```
import java.io.*;
interface In1
{
    final int a=10;
    void display();
}
```

```
class    Test    implements    Int
{
   public   void   display()
   {
     System. out. println ("Vishwas");
   }
   public   static   void   main(string [] args)
   {
      Test    t   =  new    Test();
      t. display ();
      System. out. println (a);
   }
}
```

OUTPUT:

```
Vishwas
10
```

2] Explain Abstract class with example.

-> Abstraction is used to hide the implementation details of a java program.

-> It is achieved using abstract classes.

-> For this, we must abstract keyword before any class.

-> The abstract class has some following properties.

- we cannot create instance of an abstract class

- An abstract class can contain abstract method as well as non-abstract method.

- A class must be declared abstract if it has atleast one abstract method.

- For using the abstract class, we should inherit it.

Example:

```
abstract   class   Data
{
  abstract   void   name();
}
class   value   extends   Data
{
   void name()
    {
     System.out.println("Hi! I am Vishwas");
    }
}
class   Abs
{
  public static void main (String args[])
  {
      value   v1  =  new   value();
      v1.name();
  }
}
```

OUTPUT:

```
Hi! I am Vishwas
```

3] Differentiate between interface and abstract class

Abstract class	Interface
• It can have abstract & non-abstract methods	It can have only abstract methods
• It dosen't support multiple inheritance	It supports multiple inheritance
• It can be extended using "extends" key-word	It can be implemented using "implements" keyword
• It can have class members like private, protected, public, etc.	It has its members public by default.
• It can have final, non-final, static & dynamic variables.	It can have static and final variable

• Example:
public abstract class shape {
public abstract void draw();
}
	- Example:
	public interface Drawable {
	void draw();
	}

4] Define Super keyword and its use.

-> Super keyword in java is a refrence variable which is used to refer immediate parent class object.

-> whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

-> Use's

- super can be used to refer immediate parent class instance variable
- super can be used to invoke immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

5] write an abstract class named Person and its two subclass named Student and Employee.

1. A person has a name, address, phone number, and email address.
2. A student has enrollment, course.
3. An employee has an office, salary, and designation.

Define constructors and methods for input and display for both classes. write a main program to give demonstration of all.

```
import java.util.*;
abstract class Person
{
}
class student extends Person
{
  int enrollment;
  String course;
  Student () {
      Scanner s1 = new Scanner(System.in)
     System.out.println("Enter enroll & course:");
     enrollment = s1.nextInt();
     course = s1.next();
   }
}
class Employee extends Person
{
  String office, designation;
  int salary;
  Employee() {
     Scanner s2 = new Scanner(System.in);
    System.out.println(" Enter office, designation
                         & salary: " );
    office = s2.next();
    designation = s2.next();
    salary = s2.nextInt();
  }
}
class AbstractionE2
{
 public static void main (string args[])
 {
   String name, address, emailaddress;
   int phonenumber;
   Scanner sc = new Scanner(System.in);
   Student s1 = new Student();
   Employee e1 = new Employee();
   System.out.println("Enter Name, address,
                        email & phoneno.:");
   name = sc.next();
   address = sc.next();
   emailaddress = sc.next();
   phonenumber = sc.nextInt();

   System.out.println("Name= " +name +
   "\naddress="+ address+ "\nemail= "+emaiaddress
    + "\nPhoneno= "+phonenumber + "\nEnrollment="
    + s1.enrollment + "\nCourse= " + s1.course+
    "\nOffice="+ e1.office + "\nDesignation="
     +e1.designation + "\nSalary= " + e1.salary);
 }
}
```

OUTPUT:

```
Enter enroll & course:
151240116001
IT
Enter office, designation & Salary:
AVtechs
CEO
1992000
Enter Name, address, email & phoneno.:
Vishwas
V.V.Nagar
vcharyvishwass@gmail.com
9292110110
```

Name = Vishwas
address = V.V.Nagar
email = acharyavishwass@gmail.com
Phone no = 9292110110
Enrollment = 181240116001
Course = IT
Office = AVtechs
Designation = CEO
Salary = ·199200

6] Define abstract class Shape which has three subclasses say Triangle, Rectangle and Circle. Define one Method area() in the abstract class and override this in three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle and same for other.

=>
```
import java.util.Scanner;
abstract class Shape { void area() {}
}
class Triangle {
    void area() {
        Scanner s1 = new Scanner(System.in);
        double tri, b, h, x = 0.5;
        b = s1.next Int();
        h = s1.next Int();
        tri = x * b * h;
        S.O.P(" Ans is " + tri);
    }
 }
```

```
class Rectangle {
    void area() {
        Scanner s2 = new Scanner (System.in);
        double ract, w, L;
        w = s2. next Int();
        l = s2. next Int();
        ract = w * L;
        S.O.P (" Ans is" + ract);
    }
 }
class Circle {
    void area () {
        Scanner s3 = new Scanner (System.in);
        double cir, r, pi = 3.14;
        r = s3. next Int();
        cir = pi * r * r;
        S.O.P ("Ans is " +cir);
    }
 }
class AbsArea {
    public static void main(String args[])
    {
      Triangle T1 = new Triangle();
      Rectangle R1 = new Rectangle(),
      Circle C1 = new Circle();
      Scanner sc = new Scanner (System.in);
      S.O.P (" Enter method no: ");
      int method = sc. nextInt();
      switch (method)
      {
        case 1: T1.area();
         break;
        case 2 : R1. area();
         break;
        Case 3 : c1.area();
        break;
        default: S.O.P("Enter valid input!");
        break;
      }
   }
}
```

<u>OUTPUT:</u>

```
Enter method no:
3
4
Ans is 50.24
```