□ In order to use functions/methods of math library, you need to invoke function using math keywords before the function.
e.g. x=math.abs(-7.5);
1. **pow(num1,num2)** - It computes num1 num2 , where num1 and num2 are numbers.
e.g. syste.out.print(“”+math.pow(2,3);
2. **round(num1)** - It rounds off a given number to its nearest integer. It can take float/double as argument.
e.g.
system.out.print(“”+math.round(1.5)); 2
system.out.print(“”+math.round(-1.5)); -1

Using Dates & Times in JAVA

□ Java offers two classes in java.util package to manipulate date and time.
1. java.util.Date 2. java.util.Calendar
□ In order to use Date & calendar, you need to import java.util package. E.g. import java.util.*;
Date d=new Date(); -It returns system date in the given format.
Tue Jul 20 17:30:22 GMT+05:30 2010

UNSOLVED QUESTIONS

1. What are the different types of access specifier supported by java?
2. Which is the default package of java?
3. What is friendly access of class member?
4. How does a class enforce information hiding?
5. Define an abstract class and abstract methods.
6. What is an interface? What is the use of Interface.

CHAPTER 7

CONCEPT OF INHERITANCE IN JAVA

Brief Summary of the Chapter:

This chapter talks about Inheritance, the capability of one class to derive properties from another class. Here we can learn how the inheritance is implemented in Java.

KEY POINTS

- **Inheritance:**_Inheritance is the capability of one class to inherit properties from an existing class. Inheritance supports reusability of code and is able to simulate the transitive nature of real life objects.

- **Derived/Sub and Base/Super classes**

  A class from which another class is inheriting its properties is called base class and the class inheriting properties is known as a sub class and derived class.

  1. Single (1:1)

     → when a class inherits from a single base class.

  2. Hierarchical (1:M)

     →when several classes inherit from the same class.

3. Multilevel (1:1:1)

→When a subclass is the base class of another class.

- **Method Overriding:** If Base class has a method with same signature as in sub class the method of subclass overshadows the method of base class , it is called Method overriding.
- **Method Overloading:** Two methods with same name but different signatures are there in the same scope of program.
- **Abstract Class:** The class that is used as only base class, no object of this class is used in the program.

Solved Questions:

1. What is inheritance ?

   Ans: Inheritance is a form of software reusability in which new classes are created from existing classes by absorbing their attributes and behaviours.
2. What is the primary purpose of inheritance ?

   Ans: The primary purpose of inheritance is code reusability.
3. Name three basic concepts in Java which are used in Object oriented programming.

   Ans:The three basic concepts are Classes , Objects and inheritance.
4. Which constructor is called first: subclass or superclass ?

   Ans: A subclass constructor always calls the constructor for its superclass first explicitly or implicitly to create and initialize the subclass members.
5. What is abstract class?

   Ans:An Abstract class is the one that simply represents a concept and whose objects can't be created. It is created through the use of keyword abstract.

   The superclass set up as an abstract class does not allow objects of its prototype to be created. In this case only objects of the subclass are used.
6. What is method overriding in Java?

   Ans:A method in a subclass hides or overshadows a method inherited from the superclass if both methods have same signature.(i.e. the same name, number and type of arguments and the same return type.) This property is known as Overriding the Inherited method.
7. What is an Interface in Java?

   Ans: An Interface defines a protocol of behaviour or it is a collection of constants and abstract methods. These are not classes, but they can be used in the definition of a class.
8. What is the significance of abstract class in Java program?

   Ans: When a class defines a method using the same namem return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.

   When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.
9. What types of inheritance does Java have?

   Ans: Java supports only these inheritance types:

   i. Single inheritance ii. Multilevel inheritance iii. Hierarchical
10. State True and False

    a. A subclass inherits both member variables and member methods of superclass.

    b. A class created with keyword abstract can have at the most one object.

    c. Overloading and Overriding are similar concepts in Java.

    d. Java supports single inheritance only at multiple levels of inheritance.

    e. Interfaces are used for multiple inheritance.

Ans: a. True b. False c. False d. True e. False

11. Declare and explain the basic syntax of inheritance.

Ans:The basic syntax for specifying an inherited class is as:

```
class child_class extends parent_class {
                    // class contents
                    }
```

The syntax represents the definition of the class child_class. The child_class automatically inherits an initial set of methods and variables from the parent class. The inherited variables and method can be used by the child class as if they had been declared locally in the child_class.

12. How does inheritance support software reuse?

Ans:Because a new class can be derived from an existing one, the characteristics of the parent class can be reused without the erroneous process of copying and modifying code.

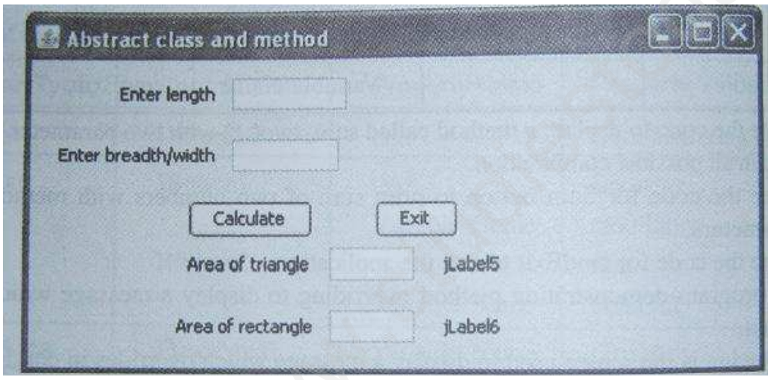
13. Differentiate between method overloading and method overriding.

Ans: Overloading: The methods with the same name but it differ by types of arguments and number of arguments.

Overriding: The methods with the same name and same number of arguments and types but one is in base class and second as in derived class. Derived class overrides over base class.

14. Write a program to find the area of triangle and rectangle through abstract class and abstract class method.

The following is the screen used to find the area of triangle and rectangle using abstract class and abstract class methods:



The list of controls for the above form is as follows:

Control Type	Control Name	Property Name	Property Value
JFrame	AbstractU1	title	Abstract class and method
JTextField	JTextFIeld1 JTextFIeld2 JTextFIeld3 JTextFIeld4	text Variable Name text Variable Name text Variable Name text Variable Name	[None] txtL [None] txtH [None] txtAreaT [None] txtAreaR
JButton	JButton1	text Variable Name text Variable Name	Calculate cmdCalc Exit cmdExit

a) Write the code to declare an abstract class Figure with an abstract method area( ). Notice that the class should declare the possible variables for area operation.

b) Write the code to declare two classes called Rectangle and Triangle which will calculate the area for both rectangle and triangle using the abstract method area( ). Use suitable constructors for both classes.

c) Write the code for Calculate button to access the abstract method for both triangle and rectangle.

d) Write the code for cmdExit to exit the application.

Ans:

a)

```
abstract class Figure {
            double L;
            double BH;
            Figure( double a, double b){
            L= a;
            BH = b;
            }
            Abstract double area( );
    }
```

b)

```
class Rectangle extends Figure {
    Rectangle ( double a, double b){
    super( a, b );
    }
    double area( ) {
            jLabel5.setText(" Inside Area of Rectangle. ");
            return L* BH;
            }
    }
    Class Triangle extends Figure {
            Traingle ( double a, double b) {
                    Super( a, b);
                    }
                    double area( ) {
                    jLabel6.setText("Inside Area for Triangle.");
                    return L * BH /2;
                    }
            }
```

c)

```
int len, heigh;
len = Integre.parseInt(txtL.getText());
heigh= Integre.parseInt(txtH.getText());
Rectangle r = new Rectangle ( len, heigh);
Triangle t = new Triangle ( len, heigh);
Figure F;
F=r;
txtAreaR.setText(String.valueOf(F.area( )));
F= t;
txtAreaT.setText(String.valueOf(F.area( )));
```

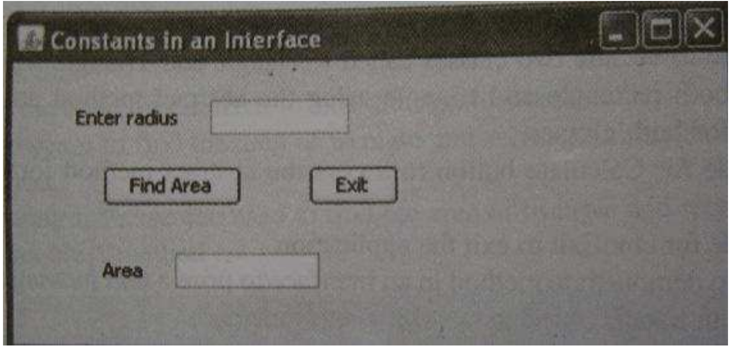
d)

```
System.exit(0);
```

15. Write a program to demonstrate constant in an interface to find the area of circle. Notice that the interface should declare the value of pie as a constant. That is:

```
 interface valuePie {
           double PIE= 3.14;
           }
```

Using class, implement the interface constant and find the area of circle. The following is the screen used to find area of circle using interface constant.:



The list of controls for the above form is as follows:

Control Type	Control Name	Property Name	Property Value
JFrame	IntAreaUi	Title	Constant in an Interface
JTextField	JTextField1 JTextField2	text Variable Name text Variable Name	[None] txtR [None] txtArea
JButton	JButton1	text Variable Name text Variable Name	Find Area cmdArea Exit cmdExit

a) Write the code to declare an interface for constant with implementation class method area().

b) Write the code for FindArea button to acecess the abstract method and display the area of circle in textbox.

c) Write the code for cmdExit to exit to exist the application.

Ans: a)

```
interface valuePie {
          double PIE=3.14;
          }
class cArea implements valuePie {
          public double area ( float r){
                    return ( PIE *r*r);
                    }
          }
```

b)

```
float r = Float.parseFloat(txtR.getText( ) );
cArea C = new cArea ( );
txtArea.setText (String.valueOf (C.area( )));
```

c)

```
System.exit(0);
```

Unsolved Questions:

1. What members of a class out of private, protected and public are inheritable?

2. When do we declare a method or class 'final'?

3. What is an abstract class?

4. When do we declare a method or class abstract?

5. What is the difference between an abstract class and an interface?

CHAPTER 8

GUI DIALOGS AND TABLES

Brief Summary of the Chapter:

This chapter tells us about how we can create dialogs in Java Swing through NetBeans IDE.
KEY POINTS:

- **A Dialog**: It is a small separate window that appears to either provide or request to/from the user.

- Java Swing provides four dialog types: **a) JDialog**(General purpose dialogs) **b) JOptionPane** (Pre-defined styles) **c) JFileChooser** (dialog for choosing files) **and d) JColorChooser**(dialog for choosing colour)

- **JOptionPane dialog Type:**

There are four built-in dialog styles:

1) Message dialog→ **JOption.ShowMessageDialog()** displays the message dialog

2) Input dialog→ **JOption.ShowInputDialog()** displays the input dialog

3) Confirm dialog→ **JOption.ShowConfirmDialog()** displays the confirm dialog

4) Option dialog→ **JOption.ShowOptionDialog()** displays the option dialog

Solved Questions:

1. What is dialog in Java?
   Ans: A dialog is a small separate window that appears to either provide or request to / from the user.
2. Write the import statement required for using JOptionPane class.
   Ans:import javax.swing.JOptionPane;
3. What is showConfirmDialog ( ) method of JOptionPane class do ?
   Ans: This method displays a dialog with several buttons and returns as int option type corresponding to the button pressed (mostly one from Yes, No or Cancel)
4. What is showInputDialog ( ) method of JOptionPane class do ?
   Ans: This method displays a dialog which is intended to receive user input, and returns a String if the input component is a text field. It displays two buttons : OK and Cancel.
5. What is the difference between a dialog created with JDialog and a dialog created with JOptionPane?