1 History introduction and language basics

Topics Covered

18. OOP Concepts
    - Object
    - Class
    - Encapsulation
    - Inheritance
    - polymorphism
19. Creating and using class with member
20. Constructor
21. finalize() method
22. static and non-static member
23. overloading
24. varargs

History of JAVA

- Java was developed by James Gosling.
- At that time it is known as “OAK”.
- But it is renamed as JAVA in 1995.
- It can follow the OOP concept.

Features of JAVA

1. **Compiled and interpreted**
- A computer language is either compiled or interpreted.
- Java combines both features therefore it is two stage system.
- First java compiler translate source code into bytecode and bytecode is not machine instruction therefore interpreter translate bytecode into machine code.

2. **Platform independent and interpreted**
- Java programs can be easily moved from one computer system to another system.
- If you upgrade in any operating system then it will not affect in the java programms.

3. **Object Oriented**
- Java is a true object-oriented language.
- Almost everything in java is an object.
- Java is an extensive set of classes which are arranged in packages and we can use it by inheritance.

4. **Robust and Secure**
- Java is a robust language because it provides many safeguards to ensure reliable code.
- It will check data type compile time and runtime.
- Security becomes an important point for a java programming on internet.

5. **Distributed**
- Java is designed as a distributed language for creating applications on networks.
- It has the ability to share both data and programs.
- In java multiple programmers at multiple remote locations to collaborate and work together on a single project.

6. Simple small and familiar

- Java is a small and simple language.
- Many features of C and C++ are not come in java so it become small.
- For example, java does not use pointers, operator overloading, preprocessor header files, goto statement, multiple inheritance.
- Java has more feature of C and C++ so that java is familiar to us.

7. Multithreaded and interactive

- Multithreaded means handling multiple tasks simultaneously.
- Java support multithreaded means we need not wait for the application to finish one task before beginning another task.
- For example, we can listen to an audio clip while scrolling a page.
- This improves interactive program.

8. High performance

- Java architecture is defined to reduce error and amount of time.
- So that java gives high performance.

9. Dynamic and Extensible

- java is a dynamic language because it is capable of dynamically linking a new class libraries, methods and classes.
- Java program support functions written in other languages such as C and C++.
- These functions are known as native methods.

10) Monitoring and Manageability

- Java supports a number of APIs, such as JVM Monitoring and Management interface.
- It manage java programs.

JDK and its Component

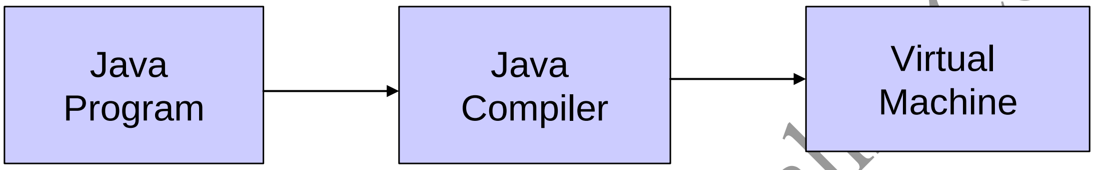
- The full form of JDK is Java Development Kit.
- The JDK comes with a collection of tools that are used for developing and running java programs.
  1) appletviewer(for viewing java applet)
  2) javac (java compiler)
  3) java (java interpreter)
  4) javap (java disassembler)
  5) javah (for C header files)
  6) javadoc (for creating HTML document)
  7) jdb (java debugger)

Tool	**Description**
Appletviewer	Enable us to run java applet.
Java	It is java interpreter which run java application and applets.
Javac	It is java compiler which translates java source code to byte code.
Javadoc	Creates HTML format documentation from java source code files.

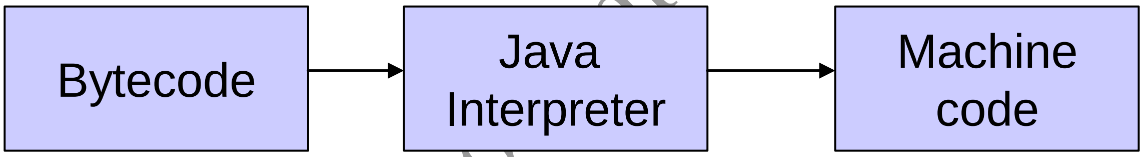
Javah	It produce header files for use with native files.
Javap	Java dissembler, which enables us to convert bytecode to source code
Jdb	Java debugger which enables to debug java program.

JVM(Java Virtual Machine)

- All language compiler translates source code into machine code for a specific computer.
- Java compiler produces an intermedia code which is known as bytecode.
- This machine is called JVM which is already inside the memory.
- JVM translate source code to byte code.



- The virtual machine code is not machine specific.
- The machine specific code is generated by the Java interpreter.
- The interpreter is different for different machine.



JRE

- The full form is Java Archive.
- A JAR file allows you to efficiently deploy a set of classes and their associated resources.
- For example a developer may build a multimedia application that uses various sound and image files.
- A set of beans can control how and when this information is presented.All of these pieces can be placed into one JAR file.
- JAR technology makes it much easier to deliever and install software.
- The elements in a JAR file are compressed, which makes downloading a JAR file much faster.

Java Editions

- there are many different editions of java

1)JAVASE

- the full form is Java Standard Edition.
- It is the normal version and it is designed for general computing.
- It has the many features of java language.
- It is compiled, object oriented and run on a virtual machine.

- Minecraft and AgroUML are the examples of JAVASE.
- It can run on a desktop application and on a web page.

2)JAVAEE

- The full form is Java Enterprise Edition.
- It has the interface specification designed to produce software that runs inside an application server implementation.
- It is designed for deployment to application server that conform to the enterprise edition servers.
- Glassfish and JBoss are the example of this.

3)JAVAME

- The full form is java Micro Edition.
- It is designed to run on mobile devices.
- It is not the same thing as android.
- The game in the mobile is example of it.

4)JAVA CARD

- It is smaller as really low end devices like smart ATM cards.

5)JAVAFX

- It is a framework designed to build rich internet client GUI application.

<u>Java IDE</u>

- It is an integrated development environment or interactive development environment.
- It is a software application that provides comprehensive facilities to computer programmer for software development.
- It consist of a source code editior, build automation tools and a debugger.

1)NetBeans

- It mainly develop for work with java but it use with other language like PHP,c/C++ and HTML.
- It is an application platform framework for java desktop application and others.
- The netbeans is written in JAVA and can run on windows,OS X,LINUX and other platform which are support JVM.
- It allows application to be developed from a set of modular software components called modules.
- It is a open source IDE.
- It support development of all java application types like JAVASE, JAVAME, JAVAFX, JAVAEE .

2)Eclipse

- It contain a base workspace and an extensible plug-in system for customizing the environment.
- Eclipse are used for various plug in system.
- It can also be used to develop application in other language like c, c++, cobol and etc.
- It can also be used to develop packages for the software mathematica.
- Eclipse is a free and open source software.

Compiling and executing JAVA program

```
class hello
{
        public static void main(String s[])
        {
                System.out.println("Hello World");
        }
}
```

Now save the program as hello.java

Compile the file using javac command

```
        Javac hello.java
```

If there is no error then hello.class file will be generated.

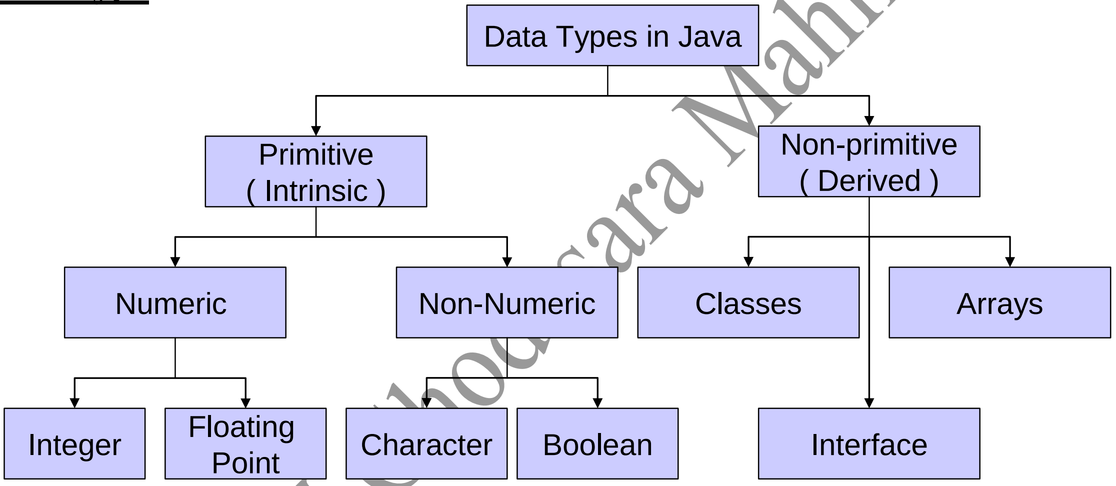
Now run the file using java command.

```
        Java hello
```

Public:it is access specifier which is public. So that main method can accessed from any class.

Static:it specifies that main() method is called without having to create object of the class.

Data type



Integer types

- Integer types can hold whole number such as 123, -97 and 6543.
- The size of the values that can be stored depends on the integer data type we choose.
- Java defines four integer types: byte, short, int and long. All of these are signed it means positive or negative values. Java does not support unsigned types.
- It should be remember that wider data types require more time for manipulation and therefore it is advisable to use smaller data types, wherever possible.
- For example, instead of storing number like 56 in an int type variable, we have to use a byte type variable to handle this number. This will improve the speed of execution of the program.

Floating point types

- Integer types can hold only whole numbers and therefore we use another type known as floating point type to hold numbers containing fractional part such as 22.43 and -2.346 (known as floating point constants).

- Floating-point numbers, also known as *real* numbers, are used when evaluating expressions that require fractional precision.
- There are two kinds of floating point storage float and double which represent single- and double-precision numbers, respectively.
- Double precision types are used when we need greater precision in storage of floating point numbers.
- All mathematical functions such as sin,cos and sqrt return double type values.

Character type

- In order to store character constants in memory, java provides a character data type called char.
- The char type assumes a size of 2 bytes but, it can hold only a single character.

Boolean type

- Boolean type is used when we want to test a particular condition during the execution of the program.
- There are only two values that a boolean type can take that is true or false.
- Boolean type is denoted by the keyword boolean and uses only one bit of storage.
- All comparison operators return boolean type values.
- Boolean values are often used in selection and iteration statements.
- The words true or false can not be used as identifiers.

Java Tokens

- The smallest individual words in a program is known as tokens.
- Following are the different types of tokens.
  1)Keywords
  2)Identifiers
  3)Literals
  4)Operators
  5)Separators

Keywords

- It is the words which meaning does not change during the program.
- Keywords combined with operators, separators according to syntax, from definition of java.

  For example
  - Class
  - Do
  - While
  - Static
  - native

Identifier

- They are used for naming classes, methods, variables, objects and packages in a program.
- There are some rules for an identifiers.
  - They can have alphabet, digits, and the underscore and dollar sign.
  - They must not begin with a digit.
  - Uppercase and lowercase are distinct.
  - They can be of any length.

Identifier must be meaningful and short.

Literal

- Literals in java are a sequence of characters (digits, letter, and other characters)that represent constant values to be stored in variables.
- Java language specifies five major types of literals.
    - Integer literals
    - Floating point literals
    - Character literals
    - String literals
    - Boolean literals

Operators

- An operator is symbol that takes one or more arguments and operates on them to produce a result.
- There are many types of operators which are available in java.
    - Arithmatic
    - relational
    - Logical
    - Increment and decrement
    - Assignment

Separators

- Seprators are symbols which are used to indicate where groups of code are divided and arranged.
- They basically defined the shape and function of our code.
- For example

    Parenthesis ( )
    Braces { }
    Brackets [ ]
    Semicolon ;
    Comma ,
    Period .

White space

- The white space can be a space, a tab or a new line.
- The extra white spaces are ignored by java compiler.
- If you write your program in a single line it makes no difference.

Comments

There are three types of comments in java.

1) **single line comment**
    - double slashes are used for single line comment.

```
//this is comment
```

2) **multi line comments**
    - the text written between /* and */ are multiline comment.

```
/* this is a
Multiline comment */
```

3) documentation comment

- this type of comment is written between /** and */

```
/**
        This is a
        Documentation comment */
```

Operators

Arithmatic operators

- arithmetic operators are used in mathematical calculations such as addition, multiplication etc.

Example

+, - , * , / , %

Relational operators

- relational operators are used to check the relation between two operands.they return a Boolean values.

Example

< , > , <= , >= , == , !=

Boolean logical

- they are used to combine two or more Boolean conditions together.
- The operands must be of Boolean type variable or conditions and the result returned is a lso a Boolean value.
  &, | , !

Bitwise logical

- Bitwise operators can be used with integer type values.
- As the name suggest they perform operations on operands bit by bit.

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise unary NOT
<<	Bitwise shift left
>>	Bitwise shift right

X	Y	X&Y	X|Y	X^Y	!X	!Y
1	1	1	1	0	0	0
1	0	0	1	1	0	1
0	1	0	1	1	1	0
0	0	0	0	0	1	1

Assignment operator

- This operator is used to assign some value or expression to a variable.
- It assigns value from the right side to the left side of expressions.

Example

int I =10;

Unary operator

- The operator which have only one operand is known as unary operator.

Example

++ , --

Shift operator

- Shift left(<<) operator shifts each bit to the left side and the shift right(>>) operator shifts each bit to the right side.

Example

A = 10 then A<<2 is shifting each bit to left side two times.

```
        A= 10        1 0 1 0
Shift left one time →  1 0 1 0 0
Shift left two time →  1 0 1 0 0  0
```

So the a <<2 = 101000 = 40

Special operator

1) **Conditional Operator(?:)**
   - This operator is known as ternary operator.

   Syntax

   Var = (condition) ? (expression1) : (expression2) ;

   Here first the condition is checked. If the condition is true then the expression1 is evaluated and its value is assigned to the var. If the condition is false then the value of expression2 is assigned to var.

2) **Instanceof operator**
   - It returns true or false value.
   - It returns true if the object written in left hand side is an object of the class written in the right hand side.
   - So it is used to check whether object is an instance of particular class or not.

   Example

   Mango instanceof fruit;

   It returns true if mango is an object of fruit class otherwise returns false.

3) **Dot operator**
   - It is used to access a member variable or method of a class through its object.

Type Casting

- Java permits mixing of constants and variables of different types in an expression.
- If byte, short and int variables are used in an expression, the result is always is in integer.
- In automatic type conversion smaller type is converted into wider type.
- In conversion, the following changes are introduced during the final assignment.

1) Float to int ignore fractional part.
2) Double to float cause rounding of digits.

Type casting means one data type is converted to another data type.
There are two type of type casting implicit and explicit.

Syntax

(type_name)Expression

Example

(int) 7.5

the answer of this is 7.

Decision statements

1)Ifelse statement

- It is used to check some condition.

Syntax:-

```
if(condition)
{
                if block
}
else
{
                else block
}
```

- If condition is true then if block executed and condition is false then else block executed.

2)Nested if else statement.

- It is used to check multiple condition.

Syntax:-

```
if(con. 1)
{
                if(con 2)
                {
                        if block
                }
                else
                {
                        else block
                }
}
else
{
                if(con 3)
                {
                        if block
                }
                else
                {
                        else block
                }
}
```

- If con 1 is true then it check for con. 2 if it is true then if block executed otherwise else block executed.
- And if con. 1 is false then it check for con. 3 if it is true then if block executed otherwise else block executed.

3)Else if ladder

- It is also used for check multiple condition.
- If con. 1 is true then block 1 executed.if it is false then it will check for con. 2 if con 2 is true then block exceuted.
- If it is false then it will check for con. 3 if it is true then block 3 executed.and if it false then else block executed.

4)Switch statement

- It is another multiple choice statement.

Syntax:-

```
Switch(variable)
{
            case 1:
                    block1;
                    break;
            case 2:
                    block 2;
                    break;
            default:
                    break;
}
```

Looping statement

1)For loop

- It is entry control loop.
- It is used for iteration.

Syntax:-

```
for(initialization;condition;increment)
{
            block;
}
```

2)While loop

- It is another entry control loop.
- It is also used for iteration.

Syntax:-

```
Initialization
while(condition)
{
            block;
            increment/decrement;
}
```

3)Do while loop

- It is an exit control loop.
- It is used for any type of iteration.

Syntax:-

```
Initialization;
do
{
            block;
            increment;
}while(condition);
```

Jumping Statements

1)Break

- The one use of break in switch statement we have seen.
- It can be used to break other loops also.
- When you use break in a loop it terminates the program.

Example

```
For(int i=1;i<100;i++)
{
        System.out.println(" i is "+i);
        If(i==5)
        {
                System.out.println(" i cannot continue");
                Break;
        }
}
```

2)Continue

- The continue statement, skips the current iteration in the loop and continues the loop to the next iteration.
- So, when we want to skip some statements based on a condition, continue statement can be used.

Example

```
For(int i=1;i<100;i++)
{
        System.out.println(" i is "+i);
        If(i%3==0||i%5==0)
        {
                Continue;
                System.out.println("no is "+i");
        }
}
```

3)Return

- The return statement causes the control to be transferred back from a method to the caller of the method.

Example

```
Int i=1;
While(i<10)
{
    If(i==3)
            Return;
    System.out.println("I = "+i);
    I++;
}
```

Array

- it is a group of variables of same data type which have a same name.
- it is used when more than one variable of same datatypes are used.

One dimensional Array

- it is a collection of variables with one row and multiple columns or one columns and multiple row.

Syntax

Data type arrayname[];

Example

Int marks[];

Here marks is the array of integer.this array is created but not ready to use.

Syntax

```
    Array_name = new data_type[size];
```

Example

```
    Marks=new int[20];
```

Both statements are combine as follow:

Int marks[]=new int[20];

We can give value as follow:

Marks[0]=55;

Int marks[]={5,6,7,8,9}

Rectangular array(two dimensional array)

- it will come in the tabular form in row or column.

Syntax

Datatype arrayname[] [] = new datatype[row][column];

Example

Int marks[][]=new int[5][5];

Int marks[][]={{1,2,3},{4,5,6},{7,8,9}};

Jagged Array

- In jagged arrays, each row, in a two-dimensional array, may contain different length.
- In this we can not only create two dimensional array but in this we can put different number of column in different row.
- For example in matrix if we create 3 row then we can put 1 column in first row, 3 column in second row and 4 column in third row.

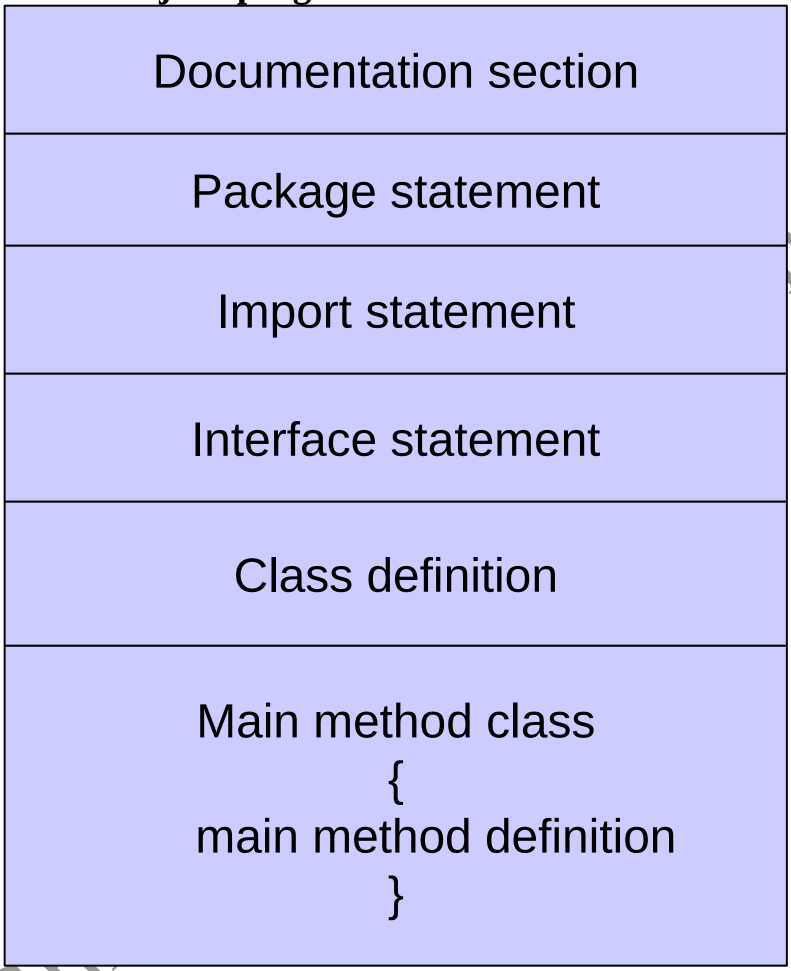
Command line argument array

- A java application can accept any number of arguments from the command line.
- This allows the user to specify configuration information when the application is launched.
- The user enters command line arguments when invoking the application and specifies them after the name of the class to be run.
- When an application is launched the run time system passes the command line argument to the application main method via an array of strings.

Example

```
Public class first
{
    public static void main(String s[])
    {
            For(String s:args)
            {
                    System.out.println(s);
            }
    }
}
Now run the program
Java First hot cold
Hot
Cold
```

Structure of java program



1)Document Section

- It is a set of comment lines which give the name of the program,the author and other details.
- We will give the comment to the line as /*…*/

2)Package statement

- Package statement is the always first statement in java program.
- It declares the package name and inform compiler that the classes defined here is belong to the package.

Example

package student;

- Here this statement is optional.and student is the package name.

3)Import statement

- This statement is come after package statement but before class definition.
- We can write number of package statements.
- This is similar to the #include statement is c.

Example

import student.test;

- This gives information to interpreter to load test class which is contained in the package student.

4)Interface statement

- An interface is same as the class but it contain the group of method declaration.
- This is also an optional statement.
- It is used when we want to implement multiple inheritance.

5)Class defination

- A java program can contain multiple class definition.
- Classes are the primary part of java program.
- The number of classes are depends on the complexity of the program.

6)Main method class

- Every java program require a main method as a starting point.
- This class is the necessary part of the program.
- A simple java program contains only this part.
- This creates the object of various classes and does communication between them.

OOP Concept

1) **Class**
   - The entire set of data, and code of an object can be made a user-defined data type with the help of a class.
   - In fact, Object are the variables of the type class.
   - Once a class has been defined , we can create number of objects belonging to that class.
   - Each object is associated with the data of type class with which they are created.
   - So, class is a collection of objects of similar type.
   - For example, mango, apple & banana are members of class fruit.
   - Classes are user-defined data types & behave like the built-in types of a programming language.
   - The syntax of creating an object is no different than the syntax used to create an integer object in c.
   - If fruit has been defined as a class, then the statement
     fruit mango;
   - It will create an object mango belonging to the class fruit.

2) **Object**
   - Objects are the basic run-time entities in the object oriented system.

- They may represent a person, a place, a table of data, a bank account or any item that a program has to handle.
- They also represent user define data.
- When a program is executed, the objects interact by sending messages to one another.
- For example if "customer" and "account" are two objects in a program then the customer object may send a message to the account object requesting for the bank balance.
- Each object contains data, code and code to manipulate that data.

3) Encapsulation

- The wrapping up of data & functions into a single unit is known as encapsulation.
- Data encapsulation is the most striking feature of a class.
- The data is not accessible to the outside world & only those functions which are wrapped in the class can access it.
- These functions provide the interface between the data of object & the program.
- This insulation of the data from direct access by the program is called data hiding or information hiding.
- Abstraction refers to the act of representing essential features without including the background details or explanations.
- Classes use the concept of abstraction & are defined as a list of abstract attributes such as size ,weight ,cost & functions to operate on these attributes.
- They encapsulate all the essential properties of the objects that are to be created.
- The attributes are sometimes called data members because they hold information.
- The functions that operate on these data are sometimes called methods or member functions.
- The classes use the concept of data abstraction , they are known as Abstract Data Types (ADT).

4) Inheritance

- Inheritance is the process by which objects of one class acquire the properties of objects of another class.
- It supports the concept of hierarchical classification.
- For example , the bird sparrow is a part of the class "**flying bird**" which is again a part of the class "**bird**".
- The principle behind this sort of division is that each derived class shares common characteristics with the class from which it is derived as illustrated in fig. :1.4.
- In OOP, the concept of inheritance provides the idea of reusability.
- This means that we can add additional features to an existing class without modifying it.
- This is possible by deriving a new class from the existing one.
- The new class will have the combined features of both the classes.
- Note , that each sub-class defines only those features that are unique to it.
- Without the use of classification , each class would have to explicitly include all of its features.

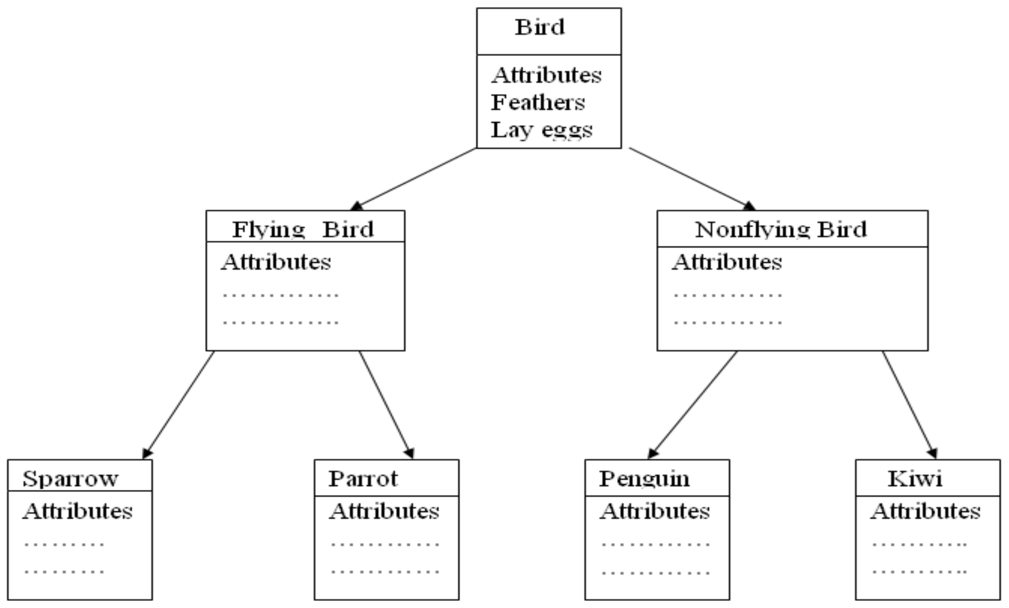
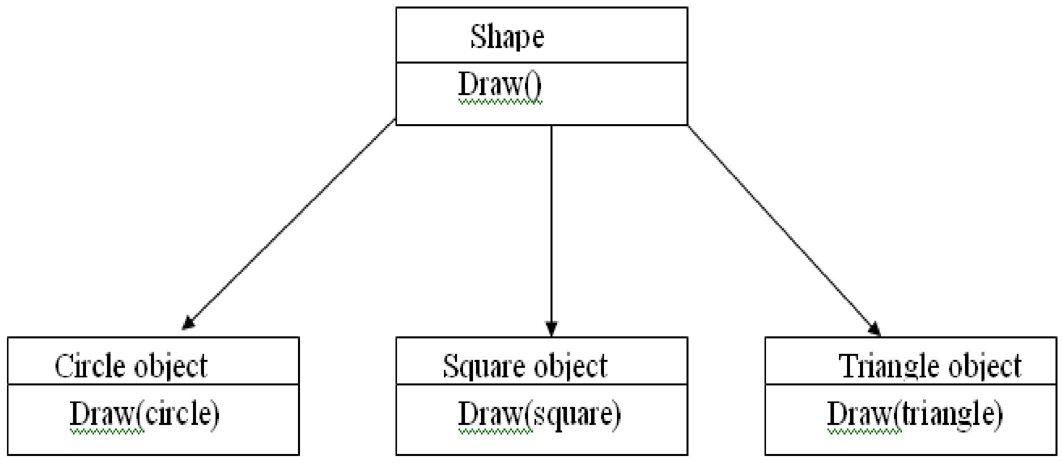


Fig. 1.4: property inheritance

5) Polymorphism

- Polymorphism , a Greek term means the ability to take more than one form.
- An operation may exhibit different behaviours in different instances.
- The behaviour depends on the types of data used in the operation.
- For example , consider the operation of addition, for two numbers, the operation will generate a sum.
- If the operands are strings then the operation would produce a third string by concatenation.
- The process of making an operator to exhibit different behaviours in different instances is known as operator overloading.
- Fig. :1.5 illustrates that a single function name can be used to handle different number & different types of arguments.



- This is something similar to a particular word having several different meanings depending on the context.
- Using a single function name to perform different types of tasks is known as function overloading.

- Polymorphism plays an important role in allowing objects having different internal structures to share the same external interface.
- This means that a general class of operations may be accessed in the same manner even through specific actions associated with each operation may differ.
- Polymorphism is extensively used in implementing inheritance.

Creating and using class

- A class is a user defined data type.
- Once the class has been defined, we can create "variables" of that type using declarations that are similar to the basic type declarations.
- In java, these variables are termed as instances of classes, which are the actual objects.
- The syntax of class definition is as follow:

```
class class name[extends superclassname]
{
        [fields declaration; ]
        [methods declaration; ]
}
```

- Everything inside the square bracket is optional.
- Classname and superclassname are valid java identifier.
- The keyword extends indicates that the properties of superclass is extended.

Examle

```
Class rect
{
        int length;
        int width;
        void getdata(int x,int y)
        {
                length=x;
                width=y;
        }
        int rectarea()
        {
                int area=length*width;
                return(area);
        }
}
```

Creating Objects

- An object in java is essentially a block of memory that contains space to store the instance variables.
- It is also known as instantiating an object.

Example

```
rectangle r1;    //declare the object
r1=new rectangle();//initiate the object
```

- The first statement declares a variable to hold the object reference and second one actually assign the objects reference to the variable.
- We can also combine both statement.

```
rectangle r1=new rectangle();
```

Constructor

- Constructor is a one type of method it's name is same as class name.
- They do not specify a return type.
- It is automatically called when the object is initialized.

Example

```
class rect
{
        int length,width;
        rect(int x,int y)
        {
                length=x;
                width=y;
        }
}
```

- We can create objects as follow:
  - Rect r1=new rect(15,10);

finalize() method

- When you will need to do some actions when an object is destroyed by garbage collection.
- For example when your program is using some non-java resources, you have to free the memory for these resources.
- This process is known as finalization.
- The finalization is done by the finalize() method.

Syntax

```
protected void finalize()
{
    //statements to be executed
}
```

- finalize() method does not return any value so the return type is void.
- The finalize() method is called just before when java runtime system performs garbage collection.

Static and non-static member

- A class has two sections. first is method declaration and second is variable declaration.
- These variables and methods are known as instance variables and instance methods.
- If we want to define a variable that is common to all the objects and access without using an object then we have to declare them as a static.

Example

```
Static int count;
Static void max(int a,int b);
```

- The variable which are declared as static are known as static variables.
- We can call static variable or static methods without using objects.
- They are also use by other class.
- Static method can only access static data.

Overloading

Method overloading

- It is possible to create methods that have the same name, but different parameter lists and different definitions.
- This is called method overloading.
- It is used when objects are required to perform similar tasks but using different input parameters.
- When we call a method in an object, java matches up the method name first and then the number of arguments and the types of arguments.

```
class rect
{
        float length,width;
        getdata(float x,float y)
        {
                length=x;
                width=y;
        }
        getdata(float a)
        {
                length=width=a;
        }
}
Rect r1=new rect();
r1.getdata(5,10)//call the first function
r1.getdata(20.5);  //call the second function
```

constructor overloading

- When we use more than one constructor in a program it is known as constructor overloading.
- In this different constructor have different number of argument or different types of argument.

```
class rect
{
                float length,width;
                rect(float x,float y)
                {
                        length=x;
                        width=y;
                }
                rect(float a)
                {
                        length=width=a;
                }
}
Rect r1=new rect(25.8,56.7);//call the first constructor
Rect r2=new rect(20.5);  //call the second constructor
```

varargs

- It means the variable arguments.
- The varargs allows the method to accept zero or multiple arguments.
- If we don't know how many argument we will have to pass in the method than varargs is the best approach.
- One best advantage is we don't have to provide overloaded methods so less code.

Syntax

Return_type method_name(data_type….variablename){}

Example

```
Class var1
{
        Static void display(String…values)
        {
                System.out.println("display method invoked");
        }
        Public static void main(String s[])
        {
                display();
                display("my","name");
        }
}
```

Rules for varargs.

- There can be only one variable argument in the method.

Variable argument must be the last argument.