

Inheritance

Class 12th

Question 1. ISC 2010

A superclass Worker has been defined to store the details of a worker. Define a subclass Wages to compute the monthly wages for the worker. The details/specifications of both the classes are given below:

Class name: Worker

Data Members/instance variables:

Name: to store the name of the worker

Basic: to store the basic pay in decimals

Member functions:

Worker (...): Parameterised constructor to assign values to the instance variables

void display (): display the worker's details

Class name: Wages

Data Members/instance variables:

hrs: stores the hours worked

rate: stores rate per hour

wage: stores the overall wage of the worker

Member functions:

Wages (...): Parameterised constructor to assign values to the instance variables of both the classes

double overtime (): Calculates and returns the overtime amount as (hours*rate)

void display (): Calculates the wage using the formula wage = overtime amount + Basic pay and displays it along with the other details

Specify the class Worker giving details of the constructor () and void display (). Using the concept of inheritance, specify the class Wages giving details of constructor (), double-overtime () and void display ()

The main () function need not be written.

```
import java.util.Scanner ;
class Worker
{
String Name;
double Basic;
public Worker (String n, double b)
{
Name = n;
Basic = b;
}
public void display ( )
{
System.out.println (Name);
System.out.println (Basic);
}
}
class Wages extends Worker
{
int hrs, rate;
double wage ;
public wage (String n, double b, int h, int r, double w)
{
super (n, b);
hrs = h;
rate = r;
wage = w;
}
public double overtime ( )
{
return (hours*rate);
}
public void display ( )
```

```

{
super.display ();
wage = overtime () + Basic;
System.out.println(wages);
}
}

```

Question 2. ISC 2011

A superclass Record has been defined to store the names and ranks of 50 students. Define a sub-class Rank to find the highest rank along with the name. The details of both classes are given below:

Class name: Record

Data members/instance variables:

name[]: to store the names of students

mk[]: to store the ranks of students

Member functions:

Record(): constructor to initialize data members

void readvalues(): to store the names and ranks

void display(): displays the names and the corresponding ranks

Class name: Rank

Data members/instance variables:

index: integer to store the index of the topmost rank

Member functions:

Rank(): constructor to invoke the base class constructor and to initialize index = 0

void highest(): finds the index/location of the topmost rank and stores it in the index without sorting the array.

void display(): displays the names and rank along with the name having the topmost rank.

Specific the class Record giving details of the constructor(), void

readvalues() and void display (). Using the concept of inheritance, specify the class Rank giving details of constructor (), void highest() and void display().

THE MAIN() FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

Question 3. ISC 2012

A superclass Detail has been defined to store the details of a customer. Define a subclass Bill to compute the monthly telephone charge of the customer as per the chart is given below:

Number of calls: Rate

1 – 100: Only rental charge

101 – 200: 60 paise per call + rental charge

201 – 300: 80 paise per call + rental charge

Above 300: 1 rupee per call + rental charge

The details of both the classes are given below:

Class name: Detail

Data members/instance variables:

name: to store the name of the customer

address: to store the address of the customer

telno: to store the phone number of the customer

rent: to store the monthly rental charge

Member functions:

Detail (...): parameterized constructor to assign values to data members

void show (): to display the details of the customer

Class name: Bill

Data members/instance variables:

n: to store the number of calls

amt: to store the amount to be paid by the customer

Member functions:

Bill (...): parameterized constructor to assign values to data members of both classes and to initialize amt = 0.0
void cal(): calculate the monthly telephone charge as per the chart is given above
void show(): displays the details of the customer and amount to be paid.
Specify the class Detail giving details of the constructor, and void show(). Using the concept of inheritance, specify the class Bill giving details of the constructor(), void cal() and void show().
THE MAIN () FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

Question 4. ISC 2013

A superclass Perimeter has been defined to calculate the perimeter of a parallelogram. Define a subclass Area to compute the area of the parallelogram by using the required data members of the superclass. The details are given below:

Class name: Perimeter

Data members/instance variables:

a: to store the length in decimal

b: to store the breadth in decimal

Member functions:

Perimeter (...): parameterized constructor to assign values to data members

double Calculate(): calculate and return the perimeter of a parallelogram is $2 * (\text{length} + \text{breadth})$

void show(): to display the data members along with the perimeter of the parallelogram

Class name: Area

Data members/instance variables:

h: to store the height in decimal

area: to store the area of the parallelogram

Member functions:

Area(...): parameterized constructor to assign values to data members of both the classes

void doarea(): compute the area as (breadth * height)

void show(): display the data members of both classes along with the area and perimeter of the parallelogram.

Specify the class Perimeter giving details of the constructor (...), double Calculate and void show (). Using the concept of inheritance, specify the class Area giving details of the constructor (...), void doarea () and void show (). The main function and algorithm need not be written.

Question 5. ISC 2014

A superclass Stock has been defined to store the details of the stock of a retail store. Define a subclass Purchase to store the details of the items purchased with the new rate and updates the stock. Some of the members of the classes are given below:

Class name: Stock

Data members/instance variables:

item: to store the name of the item

qt: to store the quantity of an item in stock

rate: to store the unit price of an item

amt: to store the net value of the item in stock

Member functions:

Stock (...): parameterized constructor to assign values to the data members

void display(): to display the stock details

Class name: Purchase

Data members/instance variables:

pqty: to store the purchased quantity

prate: to store the unit price of the purchased item

Member functions/ methods:

Purchase(...): parameterized constructor to assign values to the data members of both classes

void update (): to update stock by adding the previous quantity by the purchased quantity and replace the rate of the item if there is a difference in the purchase rate. Also, update the current stock value as (quantity * unit price)

void display(): to display the stock details before and after updation.

Specify the class Stock, giving details of the constructor() and void display(). Using the concept of inheritance, specify the class Purchase, giving details of the constructor(), void update() and void display().

The main function and algorithm need not be written.

Question 6. ISC 2015

A line on a plane can be represented by coordinates of the two-end points p_1 and p_2 as $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$.

A superclass Plane is defined to represent a line and a subclass Circle to find the length of the radius and the area of the circle by using the required data members of the superclass.

Some of the members of both classes are given below:

Class name: Plane

Data members/instance variables:

x_1 : to store the x-coordinate of the first endpoint

y_1 : to store the y-coordinate of the first endpoint

Member functions/methods:

Plane (int n_x , int n_y): parameterized constructor to assign the data members $x_1 = n_x$ and $y_1 = n_y$

void show(): to display the coordinates

Class name: Circle

Data members /instance variables:

x_2 : to store the x-coordinate of the second endpoint

y_2 : to store the y-coordinate of the second endpoint

radius: double variable to store the radius of the circle

area: double variable to store the area of the circle

Member functions/methods:

Circle(...): parameterized constructor to assign values to data members of both the classes

void findRadius(): to calculate the length of the radius using the formula:

$$((x_2 - x_1)^2 + (y_2 - y_1)^2)^{1/2} / 2$$

assuming that x_1 , x_2 , y_1 , y_2 are the coordinates of the two ends of the diameter of a circle

void findArea(): to find the area of a circle using the formula: πr^2 . The value of π is $22/7$ or 3.14

void show(): to display both the coordinates along with the length of the radius and area of the circle

Specify the class Plane giving details of the constructor and void show()

Using the concept of inheritance, specify the class Circle giving details of the constructor, void findRadius(), void find Area() and voidShow()

The main function and algorithm need not be written.

Question 7. ISC 2016

A superclass Bank has been defined to store the details of a customer. Define a sub-class Account that enables transactions for the customer with the bank. The details of both the classes are given below:

Class name: Bank

Data members/instance variables:

name: stores the name of the customer

accno: stores the account number

P: stores the principal amount in decimals

Member functions/methods:

Bank(...): parameterized constructor to assign values to the instance variables

void display (): displays the details of the customer

Class name: Account

Data member/instance variable:

amt: stores the transaction amount in decimals

Member functions/methods:

Account(...): parameterized constructor to assign values to the instance variables of both the classes

void deposit(): accepts the amount and updates the principal as $p = p + \text{amt}$

void withdraw(): accepts the amount and updates the principal as $p = p - \text{amt}$

If the withdrawal amount is more than the principal amount, then display the message "INSUFFICIENT BALANCE".

If the principal amount after withdrawal is less than 500, then a penalty is imposed

by using the formula.

$p = p - (500 - p) / 10$

void display(): displays the details of the customer

Assume that the superclass Bank has been defined.

Using the concept of Inheritance; specify the class Account giving details of the constructor(...), void deposit(), void withdraw() and void display() The superclass and the main function need not be written.

Question 8. ISC 2017

A superclass Product has been defined to store the details of a product sold by a wholesaler to a retailer. Define a subclass Sales to compute the total amount paid by the retailer with or without fine along with service tax.

Some of the members of both classes are given below:

Class name: Product

Data members/instance variables:

name: stores the name of the product

code: integer to store the product code

amount: stores the total sale amount of the product (in decimals)

Member functions/methods:

Product (String n, int c, double p): parameterized constructor to assign data

members: name = n, code = c and amount = p

void show(): displays the details of the data members

Class name: Sales

Data members/instance variables:

day: stores number of days taken to pay the sale amount

tax: to store the service tax (in decimals)

total: to store the total amount (in decimals)

Member functions/methods:

Sales(...): parameterized constructor to assign values to data members of both the classes

void compute(): calculates the service tax @ 12.4% of the actual sale amount

calculates the fine @ 2.5% of the actual sale amount only if the amount paid by the retailer to the wholesaler exceeds 30 days calculates the total amount paid by the retailer as (actual sale amount + service tax + fine)

void show(): displays the data members of the superclass and the total amount

Assume that the superclass Product has been defined. Using the concept of inheritance, specify the class Sales giving the details of the constructor (...), void

compute()) and void show(). The superclass, main function and algorithm need NOT be written.

Question 9. ISC 2018

A superclass Number is defined to calculate the factorial of a number. Define a subclass Series to find the sum of the series $S = 1! + 2! + 3! + 4! + \dots + n!$
The details of the members of both classes are given below:

Class name: Number

Data member/instance variable:

n: to store an integer number

Member functions/methods:

Number(int nn): parameterized constructor to initialize the data member n=nn

int factorial(int a): returns the factorial of a number

(factorial of $n = 1 \times 2 \times 3 \times \dots \times n$)

void display()

Class name: Series

Data member/instance variable:

sum: to store the sum of the series

Member functions/methods:

Series(...) : parameterized constructor to initialize the data members of both the classes

void calsum(): calculates the sum of the given series

void display(): displays the data members of both the classes

Assume that the superclass Number has been defined. Using the concept of inheritance, specify the class Series giving the details of the constructor(...), void calsum() and void display().

The superclass, main function and algorithm need NOT be written.

Question 10. ISC 2019

A superclass Record contains names and marks of the students in two different single dimensional arrays. Define a subclass Highest to display the names of the students obtaining the highest mark

The details of the members of both classes are given below:

Class name: Record

Data member/instance variable:

n[] : array to store names

m[]: array to store marks

size: to store the number of students

Member functions/methods:

Record(int cap): parameterized constructor to initialize the data member
size = cap
void readarray() : to enter elements in both the arrays
void display() : displays the array elements
Class name: Highest
Data member/instance variable:
ind: to store the index
Member functions/methods:
Highest(...): parameterized constructor to initialize the data members of both the classes
void find(): finds the index of the student obtaining the highest mark and assign it to 'ind'
void display(): displays the array elements along with the names and marks of the students who have obtained the highest mark

Assume that the superclass Record has been defined. Using the concept of inheritance, specify the class Highest giving the details of the constructor(...), void find() and void display().
The superclass, main function and algorithm need NOT be written.

Question 11. ISC 2020

An interface Data is defined with a data member and a method volume() which returns the volume of the implementing shape. A super class Base has been defined to contain the radius of a geometrical shape. Define a sub class CalVol which uses the properties of the interface Data and the class Base and calculates the volume of a cylinder.

The details of the members of the interface and both the classes are given below:

Interface name	Data
----------------	------

Data member:

double pi	initialize pi = 3.142
-----------	-----------------------

Interface name	Data
Member functions/methods:	
double volume()	
Class name:	Base
Data member/instance variable:	
rad	to store the radius in decimal
Member functions/methods:	
Base(...)	parameterized constructor to initialize the data member
void show()	displays the radius with an appropriate message
Class name:	CalVol
Data member/instance variable:	

Interface name	Data
ht	to store the height in decimal
Member functions/methods:	
CalVol(...)	parameterized constructor to initialize the data members of both the classes
double volume()	calculates the volume of a sphere by using the formula ($\pi \times \text{radius}^2 \times \text{height}$)
void show()	displays the data members of both the classes and the volume of the sphere with appropriate message

Assume that the interface Data and the super class Base has been defined. Using the concept of inheritance, specify the class CalVol giving the details of the constructor(...), double volume() and void show().

The interface, super class, main function, and algorithm need NOT be written.

```
class CalVol extends Base implements Data
{
    double ht;
    CalVol(double r, double h)
    {
        super(r);
    }
}
```

```

    ht=h;
}
public double volume()
{
    double x=pi * rad *rad * ht;
    return x;
}
void show()
{
    super.show();
    System.out.println("Height= " + ht);
    System.out.println("Volume= " + volume());
}
}

```

Question 12. ISC 2022

A super class **Godown** has been defined to store the details of the stock of a retail store. Define a subclass **Update** to store the details of the items purchased with the new rate and update the stock. Some of the members of both the classes are given below:

Class name : **Godown**

Data members/instance variables:

item	: to store the name of the item
qty	: to store the quantity of an item in stock
rate	: to store the unit price of an item
amt	: to store the net value of the item in stock

Member functions/methods:

Godown(...)	: parameterized constructor to assign value to the data members
void display()	: to display the stock details

Class name : **Update**

Data members/instance variables:

pur_qty : to store the purchase quantity
pur_rate : to store the unit price of the purchased item

Member functions / methods

Update(...) : parameterized constructor to assign values to the data members of both the classes

void update() : to update the stock by adding the previous quantity by the purchased quantity and replace the rate of the item if there is a difference in the purchase rate. Also update the current stock value as:
(quantity * unit price)

void display() : to display the stock details before and after updating

Assume that the super class **Godown** has been defined. Using the **concept of inheritance**, specify the class **Update** giving details of the **constructor**, **void update ()** and **void display()**.

The super class, main function and algorithm need NOT be written.