Tender Heart High School 12.08.2024

Class X Computer Applications

Topic: Class as the Basis of a all computation

Defination:

① Access Speifier : These are the terms that tells the visibility of the usage of the class members (instance variables and member methods)

② Instance variables : These variables are accessed by object. These are termed as Dynamic data members and the memory i.e allocated to these at runtime.

③ Static variables : These are also termed as class variables or copy variables. For static variables the memory is allocated at the time of compilation.

Both instance variables and static variables are declared within the class but outside the function.

④ Constructor - It is a special component of class which is used to initialize dynamic or instance data member of the class when we create object.

Evergreen
Page No.
Date : / /20

Tender Heart High School
Computer Application

Class X
Topic: Class as the Basis of all Computation

Internal Contents

① Access Specifiers
Public
private
protected
only you should know their visibility

② Data members
Static data members
- class variable
- copy variable

memory is allocated at the time of compilation

- Static int x;

these variables are writen inside the class but outside the function body with static keyword.

i.e same memory is used and sared by all the data members and objects.

dynamic data members
- instance variables
- fields of an object

memory is allocated to runtime

- int x

ie for one object one memory and for second another memory is used and allocated.

③ Member methods
Parameterized
non parameterized.

④ constructor
It is also a member method having same nameas class name.

basically used for initilizing the data members

Eg.

```
Class student
{
 int roll;
 String name;
 Student()
 {
   rollno = 0;
   name = " ";
 }
```

Q WAP by using a class with the following specifications:

class name prime

Data members : int n

Member methods :

Prime() : default constructor to initialize n

void input(x) to assign x with n.

void display() to check whether the number is prime or not.

```
class Prime
{
 int n;
 Prime()
 {
   n=0;
 }
```

datamembers

if static word is used instance variable otherwise class variable

Class variable

instance

```
void input (int xc)
{
  n=xc ;
}
void display ()
{
 int c=0 ;
 for(int i=0; i<=n; i++)
 {
  if(n%i==0)
      c++ ;
 }
 if (c==2)
   System.out.println(" Prime number");
 else
   System.out.println (" Not prime number");
 }
}
```

Local variable — Not use outside the method.

Local variable — scope is within function body

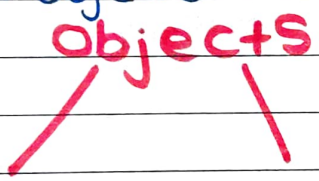
For theory part try to answer

1. What is the purpose of new operator?
2. Why is a class known as composite data type?
3. Why is a class known as user defined data type?
4. What do you mean by 'instance variables'?
5. Define instantiation.
6. Define instance of a class.
7. Define class.

Class: X Subject: COMPUTER APPLICATION
Topic: Class as the Basis of all Computation
Teacher: Prabhdeep Kaur

Good Morning Students, This lesson is for class 10 for the subject of Computer Application. Topic for today is "Class as the Basis of all Computation". In this chapter we learn about objects and classes, class as an object factory, objects as instances of a class. So let us start with this topic.

Objects: object is the fundamental unit of object oriented Programming and represents the real life entities. In OOP, the attempts are made to break a task into some components called objects



Real world objects

is one that we experience or use in our day to day life with following feature

- It is visible to us
- It has definite Shape & Size

Software objets

A Software object may be defined as an object that is created while writing a java program. The characteristics & behaviour of real world objects can be considered as data members & methods.

Camlin

Class: X Subject: Computer Application
Topic: Class as the Basis of all computation
Teacher: Prabhdeep kaur

Class class is a blue print or template for its objects that describes characteristics and behaviours. Each object of the class possesses the characteristics and behaviours desribed within the class. A number of objects of a particular class contain different characteristics but they share common behaviour. We can say that class is simply a representation of similar types of objects It is the blueprint/Plan/template that describes the details of an object. Example

```
Class Book
{
 Characteristics:
 Name
 Publisher
 Behaviours:
 Reading
 Preparing for exam
 Making project
}
```

Creating objects of a class

object is also called the instance of the class. When an object is of a class is created, then it is said to be instantiation. All the objects share the attribute and the

Camlin

Class: X Subject: Computer Application
Topic: Class as Basis of all computation
Teacher: Prabhdeep kaur

behaviour described within the class. But the attributes are unique for each object. A Single class may have any number of objects.

Syntax of creating an object of a class:

```
Book Hindi = OldBook();
```

Book → Class Name; Hindi → object Name; = → operator; OldBook() → class name; Book → class; Hindi → object

Syntax of defining a class:

```
Public class result
{
    Name
    English
    Maths
    Science

    Calculate()
    Display()
}
```

Name, English, Maths, Science → Data Members

Calculate(), Display() → member methods

Camlin

Class: X Subject: Computer Application
Topic: Class as Basis of all Computation
Teacher: Rabhdeep kaur

Class is an abstraction for the objects

Data abstraction is an act of representing essential features without including background details. When we create a class, the data items are merged or grouped with the belonging functions. To handle an object of the class, the user will only be allowed to access its functions without having direct interaction with the data members. This shows that class is an abstraction for the set of its objects.

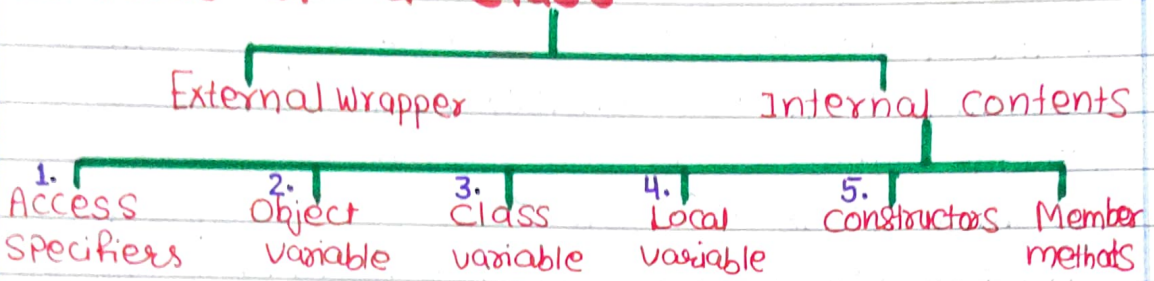
Differences between user-defined data type and primitive data type

user-defined	Primitive
1. These data types are created by the users.	1. These are built-in data types, which are provided by the system.
2. A user-defined data type includes a number of primitive types.	2. A primitive data type is independent component.

Camlin

Class: X Subject: Computer Application
Topic: Class as Basis of all Computation
Teacher: Prabhdeep kaur

Contents of a Class



* External Wrapper is the class declaration enclosing the inner part within a pair of curly brackets.

```
Class Calculate
{

}
```

The external wrapper includes following two elements:

Class: class is reserved word or keyword used to declare a class.

Class name: A class name used along with the keyword class refers by a user defined data type. Any variable used in this type of program is the object of the class.

A class name is written, the first letter as an uppercase letter and rest as lowercase. example

```
Class Worker
```

or we can use any capital letters like

```
Class WorkerPayDate
```

or underscore sign like Worker_Pay_date

Camlin

Class: X Subject: Computer Application
Topic: Class as Basis of all computation
Teacher: Prabhdeep kaur

* Internal contents of a class

1. Access Specifiers: are the terms used to specify the extent of use of the class members (variables or member methods) in the program. They are also termed as visibility modes which indicate the program area under which a class member is visible or accessible.

Access Specifiers
public — private — protected

Public: class members can even be used outside the visibility of a class.

Syntax for public declaration

data members
Public int a,b,c; (Public: specifier; int: return type; a,b,c: variable)
or
int a,b,c;

member methods
public void getdata();
(public: specifier; void: return type; getdata: method name)
or
void getdata();

When we are declaring any data member or method without specifier by default it is taken as public

Camlin

Class: X Subject: Computer Application
Topic: Class as Basis of all computation
Teacher: Prabhdeep kaur

Private: The data members or member methods which are specified as private are used only within the scope of a class. These members cannot be accessed outside the class Visibility

Syntax

data members

```
private int a,b, C;
```

member method

```
private void getdata();
```

Protected: Are used in the class as private members which can only be accessed within the class but can be used to another class during inheritance. Example

data member

```
Protected int a, b, C;
```

member method

```
Protected void getdata();
```

Break Now before going further Student let us take 5 minute break during break try to Solve Some one word questions:

1. which keyword makes class members accessible outside the class in which they are declared.
2. A class object is also known as:

Camlin

Tender Heart School

Class: X **Subject: Computer Application**

Topic: Class as Basis of all Computation

Teacher: Prabhdeep kaur

3. Name Access specifiers.
4. What are public modifiers.

Welcome back student and let us discuss the answers of the question given before going to break.

Ans 1. Public
Ans 2. Instance variable
Ans 3. Public, Private, Protected
Ans 4. A class member that is declared to be public can be accesed inside as well as outside the class.

2 Instance variable:

```
Public Class Calculate        <- Class declared
{
  int a, b, c;                <- instance variable
}
```

here variables a, b, c are the instance variable. These variables are generally used within the member methods of its class like in this calculate

Features of instance variables:

1. Instance variables are the variables declared within the class but outside the member method.

Camlin

School

Class: X Subject: Computer Application

Topic: Class as the basis of all computation

Teacher: Prabhdeepkaur

2. when a number of objects are created for the same class, the same set of instance variables are used for all.
3. Instance variables are also referred to as fields of an object.
4. Instance variables are also referred to as fields of an object.
5. Instance variables belonging to different objects contain different values.

3 Class variables is a variable that is declared within the class but outside the member methods along with the static keyword. A variable declared static will be available as a single copy to be used by all the objects of a class.

Instance variables and class variables are both declared in the same way that is outside the member methods. Only difference is that the* Instance variables are declared without Static keyword whereas, class variables are declared by using the Static keyword.* Instance variables are the individual copies of each object whereas, class variables are a common copy for each object.

Camlin

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep kaur

4. Local variables: A variable that is accessible only in a specific function or a block in which it is declared is called a local variable. Its scope is from the line of its declaration until the closing curly brace of the method within which it is declared.

Features of local variables

1. It is declared within the body of a method.
2. A local variable can never be a static variable.
3. The scope of these variables is limited within curly braces { } of a method in which they are declared.
4. They are not given initial default values. Thus you must assign a value before you use a local variable.

4 Constructors: These are the member methods which are used to initialize instance variables. They process the same name as the class name. Example

```
Class Calculate- Class
{
  int a;
  Calculate()  - constructor having same name of Class
  {
    a = ();
  }
```

Camlin

Class: X Subject: Computer Application
Topic: Basis of all computation
Teacher: Prabhdeep kaur

6. Member Methods Methods that are defined within a class are known as member methods. The member methods basically deal within the instance variables of the class. Example

```
import java.io.*;
public class Calculate
{
int m,n,s,d;
Public void inputdata()
{
 System.out.println("Enter 1st number:");
 m = in.readLine());
 System.out.println("Enter 2nd number:");
 n = in.readLine());
}
 Public void calculation()
 {
    s=m+n;
    d=m-n;
 }
 Public void output data()
 {
  System.out.println("sum" +s);
  System.out.println("difference" +d);
 }
}
```

accepts values from the user

finds the sum and difference

displays

Class : X Subject : Computer Application
Topic : Class as the basis of all computation
Teacher: Prabhdeep kaur

Features of member methods :

1. Member methods are also called member functions.
2. A member method can be declared public or private.
3. Member methods are basically used to handle class instance variables.
4. A member method may be parameterized and return the outcome.

Nested Member Methods

A member method can be used within another method to fulfill some requirements. This is known as nesting of member methods. Example

```
Class Nested
{
  int a,b,c;
  double s,ar;
  public void getdata()
  {
    a=2;
    b=3;
    c=4;
  }
  Public double perimeter()
  {
    double t = (a+b+c)/2;
    return (t);
  }
```

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep kaur

```
Public void area()
{
   S = perimeter();
  ar = Math.Sqrt(S*(S-a)*(S-b)*(S-c));
}
  Public void display()
  {
   System.out.println("Area of triangle = "+ar);
  }
   Public Static void main()
   {
     Nested ob = new Nested();
           ob.getdata();
           ob.area();
           ob.display();
      }
}
```

Perimeter() is the nested

Static Data Members

A data member declared within a class by using the static keyword is said to be a static data member. It is also known as a class variable. A static data member is a single copy available for all the objects of a class. Any change made in the static data through an abject will affect the common field available for all the objects.

Camlin

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep Kaur

Static data member is different ordinary data member of class:

* There is only one copy of this data used by the entire class which is shared by all the objects of that class.
* It is visible only within the class. It remains in the memory unless the execution of program is over.

Example:

```
Class Calculate
int a;
Static int count;
void getCount()
{ a=++count;
}
void givecount()
{ System.out.println("value of a" +a);
  System.out.println("Counter value"+count);
}
public Static void main(String args[])
 {
   Calculate ob1 = new calculate();
   Calculate ob2 = new calculate();
   Calculate ob3 = new calculate();
            ob1.getcount();
            ob2.getcount();
            ob3.getCount();
            ob1.givecount();
            ob2.givecount();
            ob3.give Count();
 }
}
```

Complin

Class: X Subject: Computer Application
Topic: Class as basis of all computation
Teacher: Prabhdeep kaur

getcount() function is called through ob1, ob2 and ob3; the value of 'a' corresponding to each object1 are assigned 1, 2 and 3 but increase in count affects the common storage field 'count'.

OutPut

value of a 1
value of count 3
value of b 2
value of count 3
value of c 3
value of count 3

Static Member method

Like static data members a member method can also be static. A static member method is a method which uses only static data members. Example

```
Class calculate
{
 int a;
 Static int count;
 void getcoun ()
 {
  a= ++count;
 }
 Static void givecount ()
 {
 System.out.println ("counter "+count);
 }
}
```

Static function uses only the data member count that declared Static in class

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep kaur

Object as Method Parameter

```
Class Xyz
{
  int a,b;
  void getval (int x, y)
  { a = x;
    b = y;
  } void sumobj (Xyz p , Xyz q)
    {
      a = p.a + q.a;
      b = p.b + q.b;
    }
    void display()
    System.out.println("Sum of a ="+a)
    System.out.println("Sum of b ="+b)
  }
  Public Static void main (String args[])
  {
    Xyz obl = new Xyz();
    Xyz ob2 = new Xyz ();
    Xyz ob3 = new Xyz();
        ob1.getval (2,3 );
        ob2.getval (4,6);
        ob3.sumobj(ob1, ob2 );
        ob3.display ();
  }}
```

Camlin

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep Kaur

In this program object ob1 and ob2 are passed to the function Sumobj (xyz p, xyz q) which are referred to p and q. In this function, variable 'a' and 'b' without object tag represent instance variable of ob3 through which function is called 'a' of ob3 contains sum of values in 'a' of objects p and q and 'b' of ob3 contains sum of values in 'b' of objects p and q. So the output is

Sum of a = 6
Sum of b = 9

Nested Class A class used within another class is known as a Nested class. Example

```
Class Biodata
{
  String name;
  int age;
  Class Addr
  {
    String city;
    String State;
    int pin;
  Public void getaddr()
    {
      City = "Chandigarh";
      State = "Chandigarh";
      Pin = "160036";
    }
```

Camlin

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep kaur

```
        Addr ad = new Addr();
    public void getbio();
      {
         name = "Sumit";
          age = 15;
      }
  }
```

Using Nested class process of invoking member methods of a nested class

```
  Class operate
  {
    Biodata ob = new Biodata();
    ob.getbio();
    ob.ad.getaddr();
```

The object of internal class is used through the object of external class to refer a member method of Internal class.

Statement ob.ad.getaddr(); The getaddr() function is called through the object ad of the internal class Addr which is in turn called through the object ob of the external class Biodata.

This keyword Sometimes it is required to use the object in a member method through which it is called. This is achieved by using this keyword. The object on which the function is called can be referred in the

Class: X Subject: Computer Application
Topic: Class as the basis of all computation
Teacher: Prabhdeep Kaur

function with this keyword. Example

```
void sum
{
  int a=5, b=8
  this.a=p.a+q.a;
  this.b=p.b+q.b;
  System.out.println(a+b)
}
```

this a and this b represent instance variable of the current object. Local variables are the variables that are declared within a function.

Camlin