

Tender Heart High School

Class: X , Computer Application

Topic: Constructors

Teacher: Prabhdeep

Good Morning Student,

This lesson is for class X for the subject of Computer Application. Today we are going to learn about constructors. We have already learnt that an object of a class contains instance variables and to perform various operations on the objects, it is required to initialize their instance variables. We can do this by using a method having same name as the class name. Such a method is called as constructor.

A constructor is a member method having same name as that of the class and is used to initialize the instance variables of the objects.

`Student student = new Student();`

This statement creates a new object called student of the student class type. The code with the new keyword looks very much like a method call. In fact, it is a call to a special kind of method called a constructor. Every class has at least one constructor method. The purpose of a constructor is to create an instance of the class. The new operator creates an empty object of the given class and then calls the constructor method, which can initialise the instance variables of the object.

Class: X, computer Application

Topic: Constructors

Teacher: Prabhdeep

Class Student — **Class name**

```
{ int rollno;  
  char grade;  
  public Student() — Constructor  
  {  
    rollno = 17;  
    grade = B;  
  }  
}
```

INVOKING A CONSTRUCTOR

A constructor can be invoked at the time of creating an object of a class. The syntax of invoking a constructor is known as shown below:

`<class name><object name> = new<constructor()>;`

Item ob = new item();

class | **operator** | **constructor**

object

Features of a Constructor

1. The constructor is defined with the same name as class name. The method Item() in above example has the same name as class name Item. Hence, it is a constructor.

Class: X , Computer Application

Topic: Constructors

Teacher: Prabhdeep

2. The Constructors is used only to initialize the data members variables and not to be used for any other task.
3. The constructor is automatically called while creating an object.
4. The constructor needs no return type.
5. The constructor is always public.
6. The constructor is overloaded automatically.

Types of Constructor

Constructors are of four types:

1. Default constructor: A constructor that is used to initialize the instance variables with default initial values is known as Default Constructor. When no constructor is defined in a program, the compiler creates a constructor on its own. As soon as an object of class is created it uses this constructor to initialize the instance variables with system approved initial values. Example

```
Class Student
{
    int rollnumber;
    char grade;
    public void Displaydata()
    {
        System.out.println("Roll no. = " + rollnumber);
        System.out.println("Grade = " + grade);
    }
    public static void main (String args[])
    {
        Student1 student1 = new Student1();
        student1.Displaydata();
    }
}
```

OUTPUT
Roll no. 0
Grade =

3

Class: X, Computer Application

Topic: Constructors

Teacher: Prabhdeep Kaur

Here, both the instance variables rollnumber and grade are printed with the default value of 0 and null, assigned by compiler.

2. Parameterised constructor: A constructor with parameters is called a parameterised constructor. The parameterised constructor allows the programmer to initialise objects with different values. This is achieved by passing the required values as arguments to the constructor method.

Example:

```
class Student
{
    int rollnumber;
    char grade;
    public Student (int roll, char gra)
    {
        System.out.println ("inside constructor");
        rollnumber = roll;
        grade = gra;
    }
    public void displaydata()
    {
        System.out.println ("Roll number=" + rollnumber);
        System.out.println ("Grade=" + grade);
    }
    public static void main (String args[])
    {
        Student student A = new Student (1, 'A');
        student A. displaydata();
        Student student B = new Student (2, 'C');
        student B. displaydata();
    }
}
```

Class: X, Computer Application

Topic: Constructors

Teacher: Prabhdeepkaur

Output
Inside constructor Roll number = 1 Grade = A
Inside constructor Roll number = 2 Grade = C

Constructor was invoked when the object was created, and it initialised the member variables with the argument values.

In parameterised constructor we must provide initial values while instantiating the object; otherwise the compiler will report an error.

3. Non Parameterised Constructor: A constructor which initialises the instance variables of an object with definite values readily available within it is known as Non-Parameterised Constructor. A non-parameterised constructor is defined with the constructor name along with the empty parameter list that is `n-const()`. Objects can be created in two ways: i. Created by compiler

Class Student

```
{ int a, b;
```

```
Student()
```

```
{ a=5; b=8;
```

```
}
```

```
void display()
```

```
{ System.out.println("value a=" + a + " & value  
b=" + b);
```

```
}
```

Output
value a = 5 & value b = 8

Class: X, Computer Application

Topic: Constructors

Teacher: Prabhdeep Kaur

The constructor Student is called automatically and the variables a and b of object ob and hri of Student class is used as a non-Parameterised Constructor.

1. A member function having the same name as that of the class name is called Constructor.
2. A constructor has _____ return type.
3. A Constructor is used when an _____ is created.
4. A constructor without any argument is known as _____.
5. A constructor is not defined with any return type. True/False

welcome back after break. Let us discuss the questions asked before going to break:

1. Constructor
2. no
3. Object
4. non-Parameterised
5. True

4. Copy Constructor: A constructor that is used to initialize the instance variables of an object by copying the initial values of

Class: X, Computer Application

Topic: Constructors

Teacher: Prabhdeep kaur

the instance variables from another object is known as copy constructor.

a. Direct entry copy constructor: In this the initial value of an object is copied by assigning it to another object. Example

```
Class Copycons
```

```
{ int a,b;
```

```
  copycons (int x, int y)
```

```
  { a = x;
```

```
    b = y;
```

```
  }
```

```
}
```

```
Class abc
```

```
{
```

```
  Public Static void main (String args [])
```

```
  {
```

```
    Copycons ob = new Copycons (5, 8);
```

```
    Copycons ob1 = ob;
```

```
  }
```

```
}
```

b). Copy constructor by passing object: In this system object is passed to the constructor. The instance variables of the current object are initialized by copying the values from object passed to the constructor.

Class: X, Computer Application

Topic: Constructor

Teacher: Prabhdeep Kaur

Some facts to be kept in mind using constructors:

1. The constructor as a member method should have the same name as the class name.
2. As a constructor is used to initialize the object's instance variables, it does not need to use return type.
3. The constructor is not called through any object but it is automatically invoked when an object is created.

Constructor overloading:

A process of using a number of constructors with the same name but different types of parameters is known as constructor overloading.

Example:

```
Class Constructor  
{
```

```
    int a, b;
```

```
    constructor ()  
    {
```

```
        a = 0 ;
```

```
        b = 0 ;
```

```
    }
```

```
    constructor (int x, int y)  
    {
```

```
        a = x ;
```

```
        b = y ;
```

```
    }
```

Class: X, Computer Application

Topic: Constructor

Teacher: Prabhdeep

```
Constructor (constructor t)
```

```
{
```

```
    a = t.a;
```

```
    b = t.b;
```

```
}
```

```
}
```

A number of constructors used in a program will have their names same as class name. Hence, they are overloaded.

Differences between Constructor and a member method

Constructor	Function/Method
1. The constructor has the same name as that of the class name.	1. A function name is different than the class name.
2. The constructor does not need to be called in the main class.	2. A method is called in the main class.
3. It will never return any value.	3. A function may or may not return a value.
4. Constructors are usually public.	4. A method may be public or private.

Unsolved programs in java based:

1. Write a program by using a class with the following specifications:

Class name : Hcf lcm

Data members/instance variables : int a,b

Member functions:

Hcf lcm(int x, int y) : constructor to initialize a=x and b=y.

void calculate() : to find and print hcf and lcm of both the numbers.

2. An electronics shop has announced a special discount on the purchase of Laptops as given below:

Category	Discount on Laptop
Up to ₹ 25,000	5.0%
₹ 25,001 – ₹ 50,000	7.5%
₹ 50,001 – ₹ 1,00,000	10.0%
More than ₹ 1,00,000	15.0%

Define a class Laptop described as follows:

Data members/instance variables:

name, price, dis, amt;

Member Methods:

- (i) A parameterised constructor to initialize the data members
- (ii) To accept the details (name of the customer and the price)
- (iii) To compute the discount
- (iv) To display the name, discount and amount to be paid after discount.

Write a main method to create an object of the class and call the member methods.

3. Write a program by using a class with the following specifications:

Class name: Calculate

Instance variables: int num, f, rev

Member Methods:

Calculate(int n): to initialize num with n, f and rev with 0 (zero)

int prime(): to return 1, if number is prime

int reverse(): to return reverse of the number

void display(): to check and print whether the number is a prime palindrome or not