 *DEPARTMENT OF INFORMATION TECHNOLOGY*

Experiment No.8

|  |  |  |
| --- | --- | --- |
| **Experiment Number** | **8** | |
| **Experiment Title** | **Method and Constructor Overloading** | |
| **Resources / Apparatus Required** | Java Developer Kit , Command Prompt | Standard PC with Windows 7,8 or 10 |
| **Objectives**  **(Skill Set / Knowledge Tested / Imparted)** | To Learn the concept of Method and Constructor Overloading. | |
| **Theory** | If a class have multiple methods by same name but different parameters, it is known as **Method Overloading**.  If we have to perform only one operation, having same name of the methods increases the readability of the program.   |  | | --- | | There are three ways to overload the method in java |  1. By changing number of arguments 2. By changing the data type 3. By changing the order of arguments   Like methods, constructors can also be overloaded. | |
| **Program & output** | 1. **Method Overloading**   import java.lang.\* ;  import java.util.\* ;  import java.math.\* ;  class Area  {  public double area(double r)  {  return(3.14\*r\*r);  }    public double area(double l,double b)  {  return(l\*b);  }    public double area(double a,double b,double c)  {    double s=(a+b+c)/2;  s=s\*(s-a)\*(s-b)\*(s-c);  return(Math.sqrt(s));  }  }  class UseArea  {  public static void main(String []args)  {  Area a =new Area();  System.out.println("Area of Circle = "+a.area(10));  System.out.println("Area of Rectangle = "+a.area(10,10));  System.out.println("Area of Triangle = "+a.area(10,10,10));  }  }   1. **Constructor Overloading**   import java.lang.\* ;  import java.util.\* ;  import java.math.\* ;  class Box  {  private int l;  private int b;  private int h;    public Box()  {  Scanner sc =new Scanner(System.in);  l=sc.nextInt();  b=sc.nextInt();  h=sc.nextInt();  }    public Box(int s)  {  l=b=h=s;  }    public Box(int i,int j,int k)  {  l=i;  b=j;  h=k;  }  public Box(Box p)  {  l=p.l;  b=p.b;  h=p.h;  }  public void show()  {  System.out.println("Lenght = "+l+" Breadth = "+b+" and Height = "+h);  }  }  class UseBox  {  public static void main(String []args)  {  Box b1,b2,b3,b4;  b1=new Box();  b2=new Box(15);  b3=new Box(5,10,15);  b4=new Box(b3);  b1.show();  b2.show();  b3.show();  b4.show();  }  }  **C:\Users\Ashu\Desktop\Coding\Codes\Java Programs\Screen Shots\UseBox.png** | |
| **Conclusion** | Hence nested loops can be used to execute various programs. | |