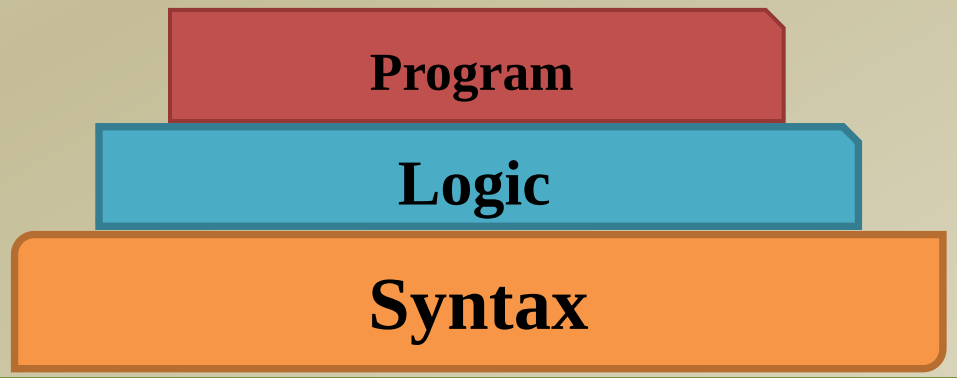
Code at Random (OPC) PVT. LTD.

Function Overloading



Function Overloading

- When two or more function have same name but different signatures, they are called overloaded functions and the process of creating and calling their overloaded methods is called function overloading.
- It implements polymorphism.
- **Function Signature :-** Function name along with parameter list is called function signature. It is used in differentiating between two functions when they have same name.
- **Representation:** `void Func1 (int a)`

  Return Type → Function Signature
- Function Signature can be changed by changing:
  - *Number Of Parameters*
  - *Type Of Parameters*
  - *Order Of Parameters*
- Note:- There is no role of return type in function signature and also in function overloading.

Programs Based on Overloading

1. WAP to create overloaded functions:
   - *void sumseries(int n, double x)*
     to print sum of series:
     $S = x - \frac{x}{2} + \frac{x}{3} - \ldots$ ***n terms***
   - *void sumseries()*
     to print sum of series:
     S = 1! + 2! + 3! + … + 20!
   - *Also create main() function and call the given functions*

2. WAP to overload functions:
   - void polygon(int n, char ch)
     draws a filled square of side n using character ch
   - void polygon(int x, int y)
     draws a rectangle using character @
   - void polygon() to display
     *
     **
     ***
   - *Also create main() function and call the given functions*

1.
```
import java .util.*;
class overload1{
public static void main(){
    int n;
    double x;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the n and x");
    n=sc.nextInt();
    x=sc.nextInt();
    sumseries(n,x);
    sumseries();
}
static void sumseries(int n, double x){
    double s=0;
    int sign=1;
    for(int i = 1; i<=n;i++){
        s=s+(x/i)*sign;
        sign*=-1;
    }
    System.out.println(s);
}
static void sumseries(){
    double s=0,f;
    for(int i = 1;i<=20;i++) {
        f=1;
        for(int j=1;j<=i;j++)
            f=f*j;
        s=s+f;
    }
    System.out.println(s);
}
}
```

2.
```
class overload2{
public static void main(){
    overload2 ob = new overload2();
    ob.polygon(3,'&');
    ob.polygon(3,6);
    ob.polygon();
}
void polygon(int n, char ch){
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n;j++){
            System.out.println(ch);
        }
    }
}
void polygon(int x, int y){
    for(int i=1; i<=x;i++){
        for(int j=1;j<=y;j++){
            System.out.println("@");
        }
    }
}
void polygon(){
    for(int i=1;i<=3;i++){
        for(int j=1;j<=i;j++){
            System.out.println("*");
        }
    }
}
}
```

Programs Based on Overloading

3. WAP to overload functions:

- *void num_calc(int n, char ch)*
  computes square of the integer if ch is s otherwise compute its cube.
- *void num_calc(int a, int b, char ch)*
  computes product if ch is p otherwise computes its sum.
- *void num_calc(char a, char b)*
  checks the equality of two characters

4. WAP to overload functions:

- *void compare(int a, int b)*
  to print greater value
- *void compare(char a, char b)*
  to print character with higher numeric value
- *Also create main() function and call the given functions*

3.
```
class overload3{
    public static void main(){
        overload3 ob = new overload3();
        ob.num_calc(3,'&');
        ob.num_calc(3,6,'$');
        ob.num_calc('%','(');
    }
    void num_calc(int n, char ch){
      if(ch=='s')
        System.out.println(n*n);
      else
        System.out.println(n*n*n);
    }
    void num_calc(int x, int y, char ch){
        if(ch=='p')
         System.out.println(x*y);
      else
        System.out.println(x+y);
    }
    void num_calc(char a, char b){
        if(a==b)
          System.out.println("TRUE");
        else
          System.out.println("FALSE");
    }
}
```

4.
```
class overload4{
    public static void main(){
        overload4 ob = new overload4();
        ob.compare(3,7);
        ob.compare('#','$');
    }
    void compare(int a, int b){
      if(a>b)
        System.out.println(a);
      else
        System.out.println(b);
    }
    void compare(char a, char b){
        if(a>b)
         System.out.println(a);
      else
        System.out.println(b);
    }
}
```

Programs Based on Overloading

5. WAP to overload functions:
   - *double area(double a, double b, double c)*
     returns area of scalene triangle
   - *double area(int a, int b, int h)*
     returns area of trapezium
   - *double area(double d1, double d2)*
     returns area of rhombus
   - *Also create main() function and call the given functions*

6. WAP to overload functions:
   - double series(int n)
     to print the sum of series:
     S = $1 + \frac{1}{2} + \frac{1}{3} + \ldots$ n terms
   - double series(double a, double n)
     to print the sum of series:
     S = $\frac{1}{a^2} + \frac{4}{a^5} + \frac{7}{a^8} + \cdots n\ terms$
   - *Also create main() function and call the given functions*

5.
```
class overload5{
    public static void main(){
        overload5 ob = new overload5();
        ob.area(3.5,8.9,9.8);
        ob.area(3,7,9);
        ob.area(5.8,9.7);
    }
    double area(int a, int b,int c){
       double area = ((a+b)*c)/2.0;
       return area;
    }
    double area(double a,double b, double c){
        double s= (a+b+c)/2.0;
        double area = Math.sqrt(s*(s-a)*(s-b)*(s-c));
        return area;
    }
    double area(double d1, double d2){
        double area= (d1*d2)/2.0;
        return area;
    }
}
```

6.
```
class overload6{
    public static void main(){
        overload6 ob = new overload6();
        ob.series(10);
        ob.series(2.0,6.0);
    }
    double series(int n){
       double s=0.0;
       for(int i=1;i<=n;i++){
           s=s+(1/i);
       }
       return s;

    }
    double series(double a,double n){
        double s= 0.0;
        for(int i=1;i<=n;i+=3){
            s=s+(i/Math.pow(a,i+1));
        }
        return s;
    }
}
```