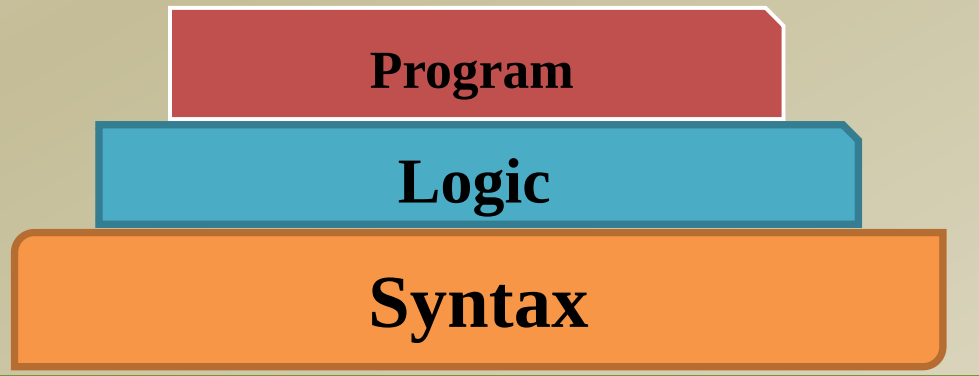
Code at Random (OPC) PVT. LTD.

Programming with Parameters



Programming Using Parameters

1. WAP to create function boolean isPalin(int n) & char isPrime(int n) & print the PalPrime numbers between 1 to 1000.
2. WAP to create function int isPerf(int n) & char isArm(int n) & print the PerfArm numbers between 1 to 1000.
3. WAP to create function int isArm(int n) & print the Armstrong numbers between 1 to 100.
4. WAP to create functions:
   void power(int a, int b)
   void add(int a, int b)
   void sub(int a, int b)
   void prod(int a, int b)
   void div(int a, int b)
   void mod(int a, int b)

   and call all of them from main() function.

   Note:- Perform the operation in the function as given without using their specific operator.

Boolean isPalin() and char isPrime ()

```
class palprime {
   boolean isPalin(int n) {
      int d,r=0,t;
      t=n;
      while(t>0) {
         d= t%10;
         r= r*10+d;
         n=n/10;
      }
      if(r==n)
         return true;
      else
         return false;
    }
   char isPrime(int n) {
      int c=0;
      int i;
      for(i=1;i<=n;i++) {
         if(n%i==0)
            c++;
       }
       if(c==2)
         return 'y';
       else
         return 'n';
      }
    public static void main(String Args[])  {
      palprime ob = new palprime();
      boolean x;
      char a;
      for(int i= 1; i<=10; i++)  {
         x= ob.isPalin(i);
         a= ob.isPrime(i);
         if(x==true && a=='y')
           System.out.println(i);
      }
     }
}
```

Int isPerf() and char isArm ()

```
class perf_arm {
    int isPerf(int n)  {
        int i,s=0;
        for(i=1;i<=n;i++) {
          if(n%i==0)
            s=s+i;
        }
        if(s==n)
          return 1;
        else
          return 0;
 }
char isArm(int n) {
        int d,t,s=0;
        t=n;
        while(t>0) {
            d= n%10;
            s= s+(d*d*d);
            t= t/10;
        }
        if(s==n)
          return 'Y';
        else
          return 'N';
}
    public static void main(String Args[])  {
     perf_arm ob = new perf_arm();
     int x;
     char a;
     for(int i= 1; i<=1000; i++)  {
         x= ob.isPerf(i);
         a= ob.isArm(i);
         if(x==1 && a=='Y')
            System.out.println(i);
     }
    }
}
```

Int isArm( )

```
class armstrong {
   int  isArm(int n) {
        int d,t,s=0;
        t=n;
        while(t>0) {
            d= n%10;
            s= s+(d*d*d);
            t= t/10;
        }
     if(s==n)
      return 1;
    else
      return 0;
 }
```

```
public static void main(String Args[])  {
    armstrong ob = new armstrong();
     int x;
     for(int i= 1; i<=100; i++)  {
         a= ob.isArm(i);
         if(a= =1)
            System.out.println(i);
     }
    }
}
```

Program 4 - Operations

```
import java.util.*;
class program4 {
    void power(int a, int b)  {
        int i,p=1;
        for(i=1;i<=b;i++)
            p= p*a;
       System.out.println(p);
    }
    void add(int a, int b)  {
        int i,s=a;
        for(i=1;i<=b;i++)
          s++;
       System.out.println(s);
    }
    void subtract(int a, int b) {
        int i, s=a;
        for(i=1; i<=b;i++)
            s--;
      System.out.println(s);
    }
    void product(int a , int b) {
        int i, p=0;
        for(i=1;i<=b;i++)
            p=p+a;
        System.out.println(p);
      }
     void divide(int a, int b) {
      int max= a<b?b:a;
      int min= a<b?a:b;
      int c=0,r;
      r= max;
      while(r>=min) {
          c++;
          r= r-min;
      }
       System.out.println(c);
    }
    void mod(int a, int b) {
      int max= a<b?b:a;
      int min= a<b?a:b;
      int r;
      r= max;
        while(r>=min)
            r= r-min;
         System.out.println(r);
     }
    public static void main(String Args[])
        {
            Scanner sc= new Scanner(System.in);
            int a,b;
            System.out.println("Enter two numbers");
            a= sc.nextInt();
            b= sc.nextInt();
            program4 ob= new program4();
            ob.power(a,b);
            ob.mod(21,4);
            ob.divide(67,8);
            ob.product(16,19);
            ob.mod(67,9);
        }
    }
```