I/O: `Scanner in=new Scanner(System.in); int n=in.nextInt(); String s=in.nextLine(); System.out.printf("%.2f %8.1f %-10s\n",d,d,s);` **Control:** `if(c){}else{}` `for(int i=0;i<n;i++){}` `while(c){}` `switch(x){case 1:stmt;break;default:stmt;}` **Method:** `public static int m(int x) throws Exception{if(x<0)throw new IllegalArgumentException();return x*2;}` **Arrays:** Min:`int min=a[0];for(int i=1;i<a.length;i++)if(a[i]<min)min=a[i];` Max:`int max=a[0];for(int i=1;i<a.length;i++)if(a[i]>max)max=a[i];` 2nd:`int max=a[0],sec=a[0];for(int x:a)if(x>max){sec=max;max=x;}else if(x>sec&&x!=max)sec=x;` Sum:`double sum=0;for(int x:a)sum+=x;` Search:`int idx=-1;for(int i=0;i<a.length;i++)if(a[i]==t){idx=i;break;}` Count:`int c=0;for(int x:a)if(x==t)c++;` Copy:`int[]cp=new int[a.length];for(int i=0;i<a.length;i++)cp[i]=a[i];` **Class:**

```java
public class Student {
  private String name; private int id; private double[] grades; private static int count=0;
  public Student(){name="";id=0;grades=new double[0];count++;}
  public Student(String n,int i,double[]g){name=n;id=i;grades=new double[g.length];
    for(int j=0;j<g.length;j++)grades[j]=g[j];count++;}
  public Student(Student s){name=s.name;id=s.id;grades=new double[s.grades.length];
    for(int i=0;i<s.grades.length;i++)grades[i]=s.grades[i];count++;}
  public String getName(){return name;}
  public double[]getGrades(){double[]c=new double[grades.length];for(int i=0;i<grades.length;i+
  public void setName(String n){if(n==null)throw new IllegalArgumentException();name=n;}
  public double getGPA(){double sum=0;for(double g:grades)sum+=g;return sum/grades.length;}
  public static int getCount(){return count;}
  public boolean equals(Student s){return name.equals(s.name)&&id==s.id;}
  public String toString(){return name+"("+id+")GPA:"+getGPA();}
}
```

Exception: `try{int x=Integer.parseInt(s);}catch(NumberFormatException e){x=0;}` **Validation:** `boolean v=false;int n=0;while(!v){try{n=in.nextInt();if(n>=1&&n<=100)v=true;}catch(InputMismatchException e){in.nextLine();}}` **File:** Read:`Scanner f=new Scanner(new File("x.txt"));while(f.hasNext())System.out.println(f.nextLine());f.close();` Write:`PrintWriter p=new PrintWriter("x.txt");p.println("text");p.close();` Until:`Scanner f=new Scanner(new File("n.txt"));int c=0;while(f.hasNextInt()){int n=f.nextInt();if(n<0)break;c++;}f.close();` **Random:** `Random r=new Random();int x=r.nextInt(100)+1;int range=r.nextInt(max-min+1)+min;` **ArrayList:** `ArrayList<Integer>list=new ArrayList<>();list.add(5);list.get(0);list.set(0,10);list.remove(0);list.size();` **String:** `s.length()` `s.charAt(0)` `s.substring(1,3)` `s.equals("x")` `s.indexOf("x")` Count:`int c=0;for(int i=0;i<s.length();i++)if(s.charAt(i)=='x')c++;` **Common Classes:** Rectangle:`private double l,w;public Rectangle(){this(1,1);}public Rectangle(double x,double y){setL(x);setW(y);}public void setL(double x){if(x<=0)throw new IllegalArgumentException();l=x;}public double getArea(){return l*w;}` Box:`private double l,w,h;public double getVolume(){return l*w*h;}` BankAccount:`private double bal;public void deposit(double a){if(a<=0)throw new IllegalArgumentException();bal+=a;}public void withdraw(double a){if(a>bal)throw new IllegalArgumentException();bal-=a;}` **Object Arrays:** Best:`Student best=arr[0];for(Student s:arr)if(s.getGPA()>best.getGPA())best=s;` Count:`int c=0;for(Student s:arr)if(s.getGPA()>=3.5)c++;` Filter:`ArrayList<Student>list=new ArrayList<>();for(Student s:arr)if(s.getGPA()>=3.0)list.add(s);` **Loops:** Nested:`for(int i=1;i<=10;i++){for(int j=1;j<=10;j++)System.out.printf("%4d",i*j);System.out.println();}` Sentinel:`int sum=0,n;while((n=in.nextInt())!=-1)sum+=n;` Flag:`boolean found=false;for(int i=0;i<arr.length&&!found;i++)if(arr[i]==target)found=true;` **Quick Ref:** Area=l×w, Vol=l×w×h, C→F=C×9/5+32 **Tips:** Deep copy arrays, static→static only, close files, clear scanner buffer **Avoid:** int division(use 9.0/5), ==for strings(use .equals), off-by-one, shallow copy