

```
| poslyrel>  
| confibras>  
| rastundeas>  
| poctonal) sterie;  
  
| dectamc> ist;  
| osttude>>  
| decprelnc:al ter;  
| cocypinc <satinf;  
  
| cecrimdears>  
| custanatisglass; ← 1 ( ← 7 3 1 7 ) )  
  
| compatrless>;  
| ontunalecis>;  
| dherdilgessstrte;  
| dutiresst><kl;  
| deconaitecteptls;  
| costinlate;
```

# Loops, Conditions, and Parsing

Fundamental programming concepts for creating efficient and intelligent software.



by **Joseph Nishimwe**



# Core Concepts

## 1 Loops

Repeat actions multiple times.

## 2

## Conditions

Help programs make decisions.

## 3 Parsing

Interpret data for program use.

# Making Computers Useful

## Efficiency

Programs repeat tasks and make decisions.

## Real-World

Apps use loops and conditions daily.



# Improving Efficiency and Automation



## Automation

Automate repetitive tasks.



## Time Savings

Reduce time and effort.







# Making Data Usable

## Data Interpretation

Interpret user input.

## Decision Making

Use data to make decisions.

## Relevant Content

Show relevant content.



# Helping Software Work Smarter

1

## Smart Decisions

Programs make smart decisions.

2

## Customization

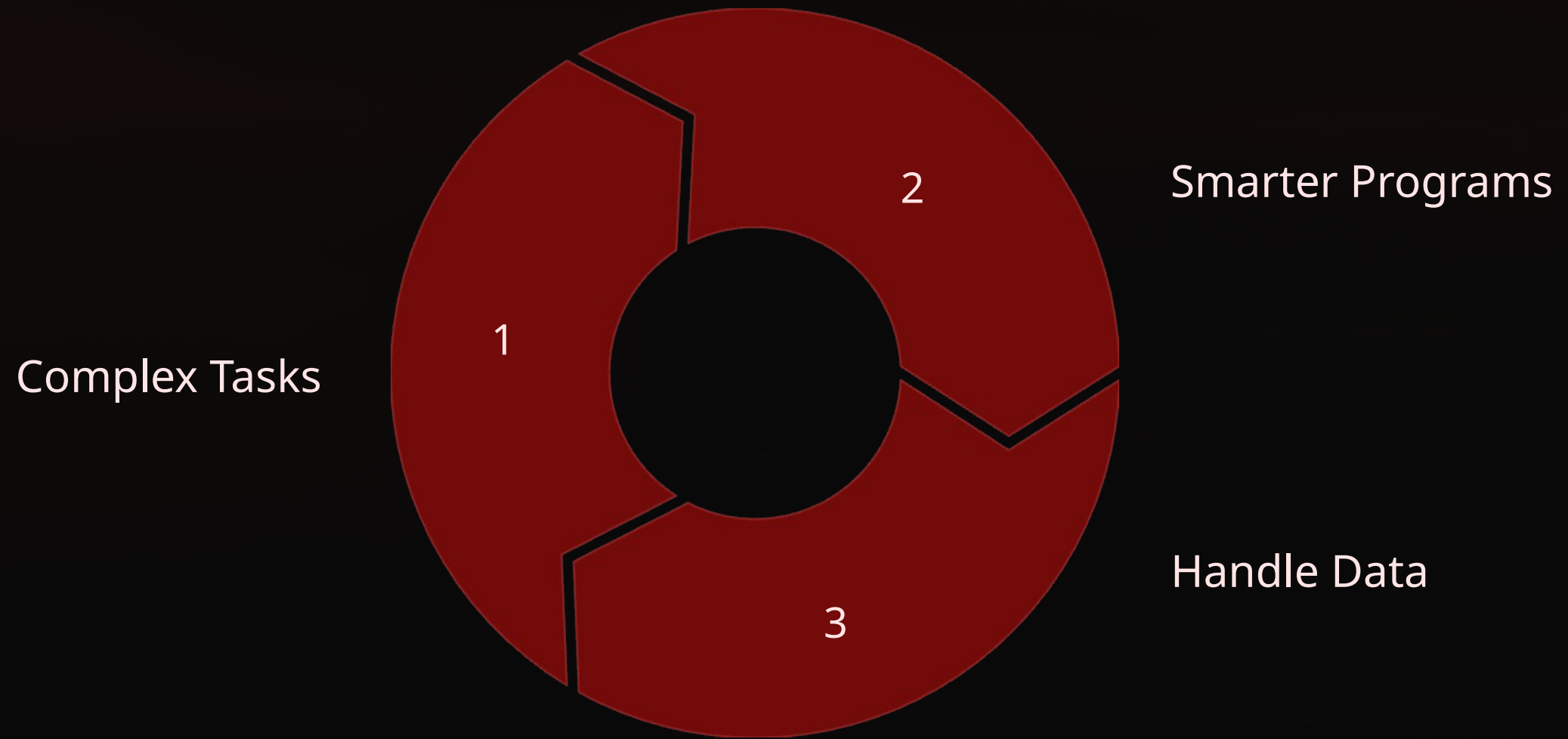
Show relevant products.

3

## Responsiveness

Software is more helpful.

# Improving Program Functionality





A stylized illustration of a forest scene. A river flows through the center, with a bridge crossing it in the middle ground. A small house with a red roof is situated on the right bank. The background is filled with tall, dark evergreen trees. The overall color palette is dominated by dark blues, greys, and a vibrant red for the house and some foliage.

# Managing Data Effectively

1

## Data Interpretation

Essential for interpreting data.

2

## Data Handling

Poor handling leads to mistakes.

3

## Reliable Apps

Important for building reliable apps.



# Conclusion

Loops, conditions, and parsing are foundational skills.

Key for efficient, responsive, and data-driven software.

