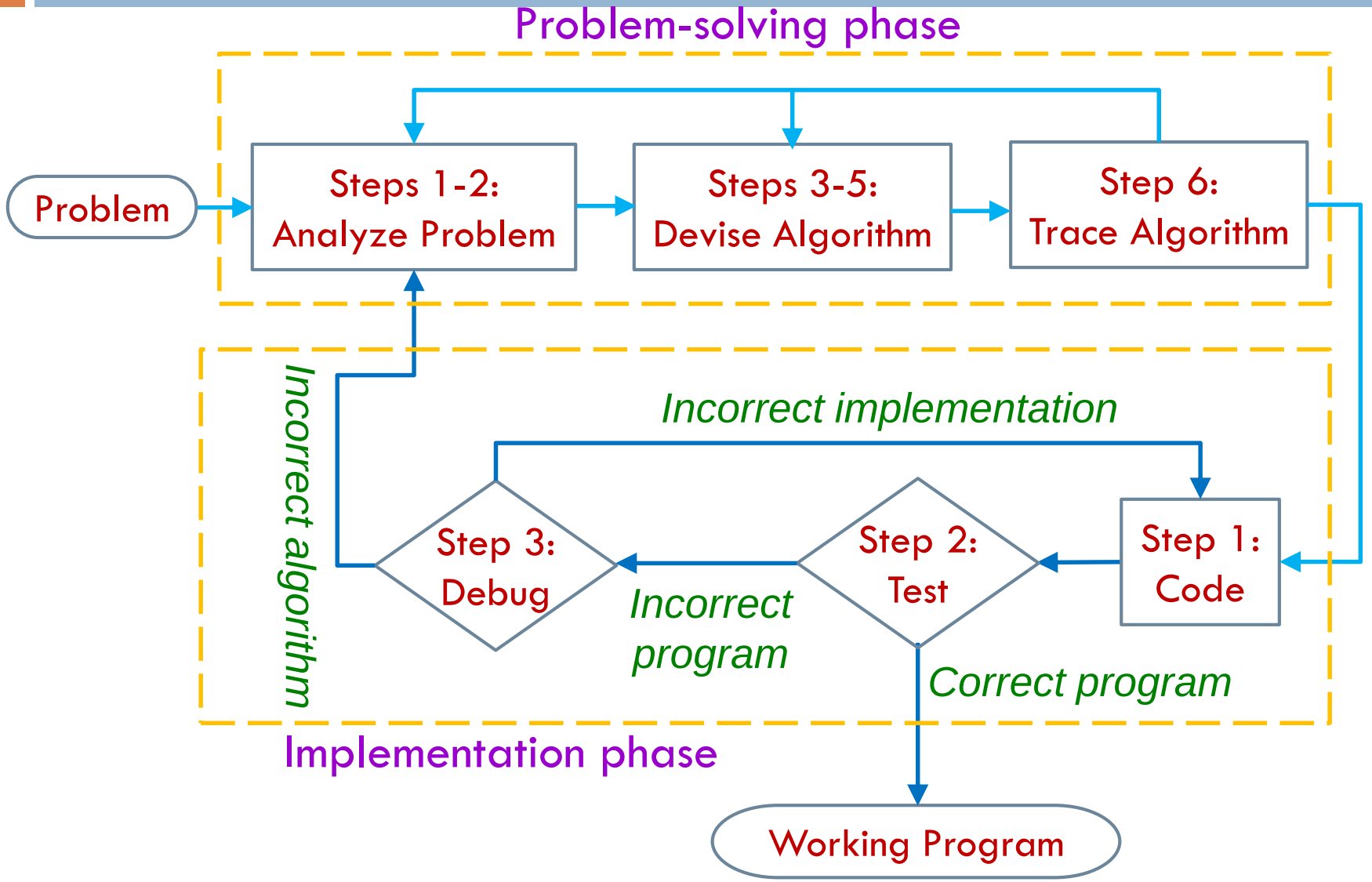
HIGH-LEVEL PROGRAMMING I

Program Development Process by Prasanna Ghali

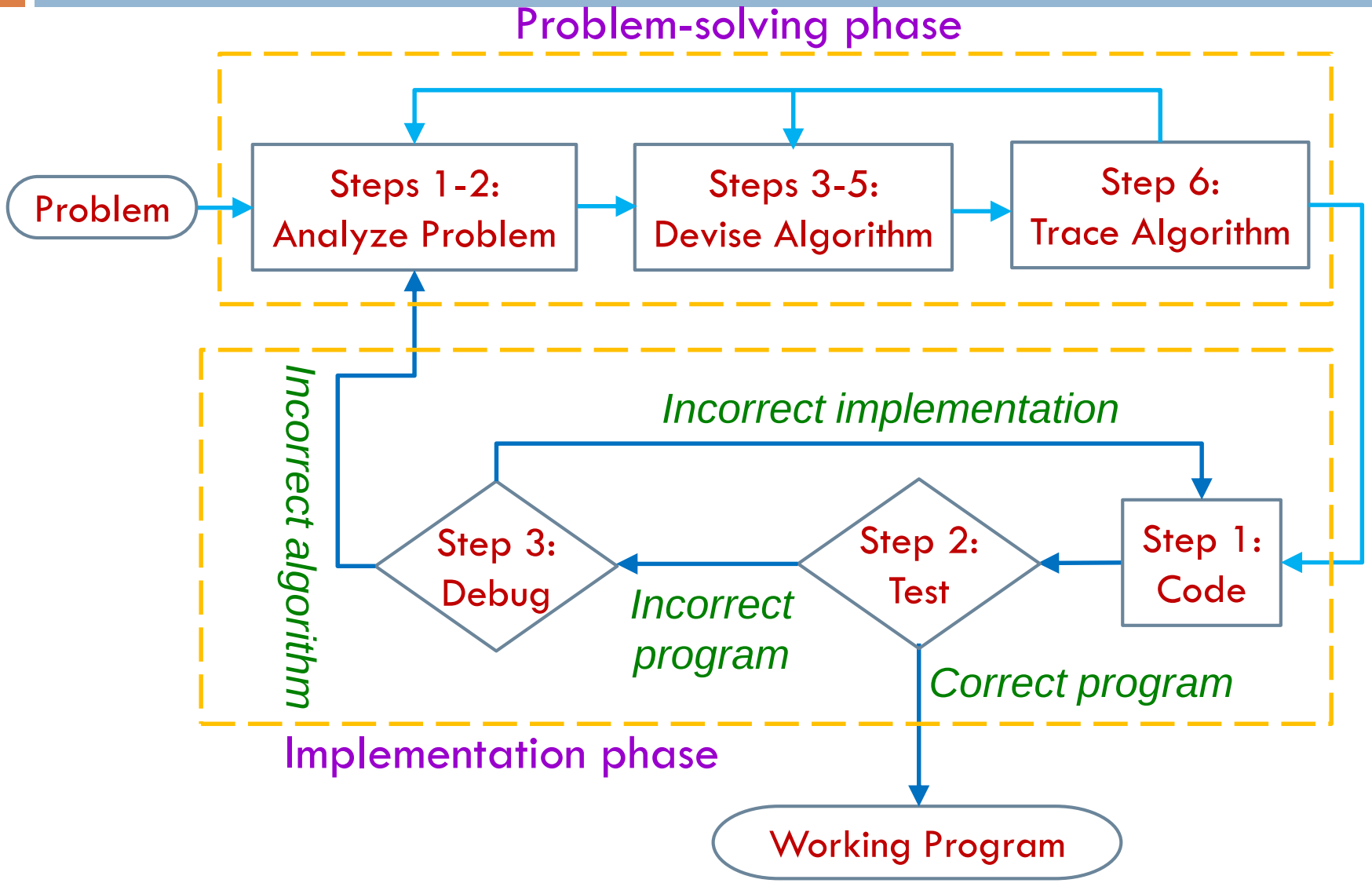
Outline

- Program development process
- Six steps of problem-solving phase
- Implementation phase

Program Development Process

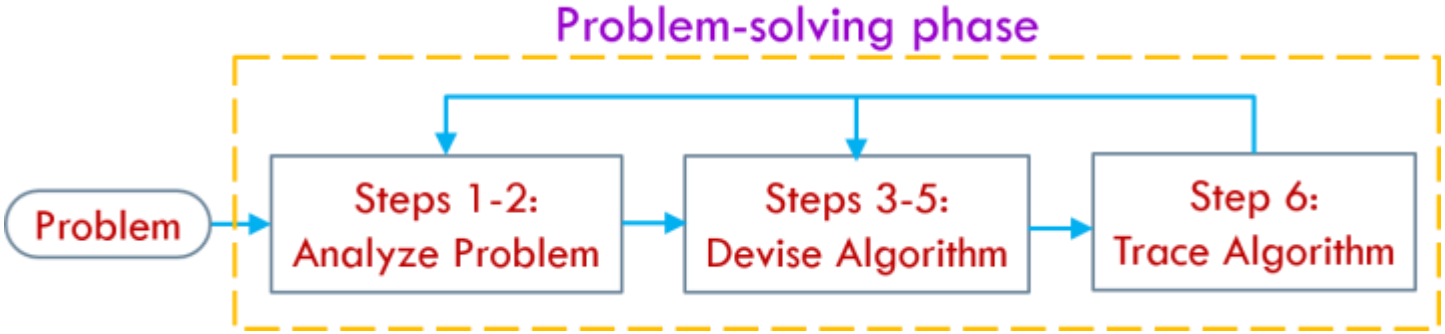


Program Development Process



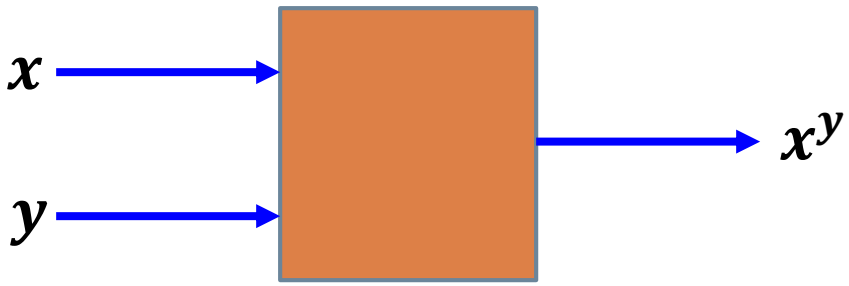
Problem-Solving Phase

1. Understand problem clearly
2. Describe inputs and outputs
3. Work problem by hand for simple data set
4. Decompose solution into step-by-step details
5. Generalize steps into algorithm
6. Test algorithm with broader variety of data



Algorithm for x^y (1/8)

- Step 1: Understand problem clearly
  - If $x = 2, y = 3$, then $2^3 = 8$
  - If $x = 3, y = 4$, then $3^4 = 81$
- Step 2: Describe input and output clearly
  - Inputs are *positive* integer values, output is integer value



Algorithm for x^y (2/8)

- Step 3: Work the problem by hand for simple data set
  - Set $x = 3, y = 4$
  - Multiply 3 by 3
    - You get 9
  - Multiply 3 by 9
    - You get 27
  - Multiply 3 by 27
    - You get 81
  - 3^4 is 81

Algorithm for x^y (3/8)

- Step 4: Decompose solution into step-by-step details
  - Each step must be precise
  - Nothing is left to guesswork

Algorithm for x^y (4/8)

- Step 5: Generalize steps into algorithm – you'll need to see underlying pattern to solve problem
- Requires two activities:
  - Replace particular values used in each step with mathematical expressions of parameters
  - Find repetition in terms of parameters

Algorithm for x^y (5/8)

- Replace particular values used in each step with mathematical expressions of parameters
  - Set $x = n = 3, y = 4$
  - Multiply x by $n = 3$
    - You get $n = 9$
  - Multiply x by $n = 9$
    - You get $n = 27$
  - Multiply x by $n = 27$
    - You get $n = 81$
  - x^y is n

Algorithm for x^y (6/8)

- Find repetition
  - Set $n = x = 3, y = 4$
  - $n = multiply\ x\ by\ n$
  - $n = multiply\ x\ by\ n$
  - $n = multiply\ x\ by\ n$
  - $n = x^y$ is $result$

Algorithm for x^y (7/8)

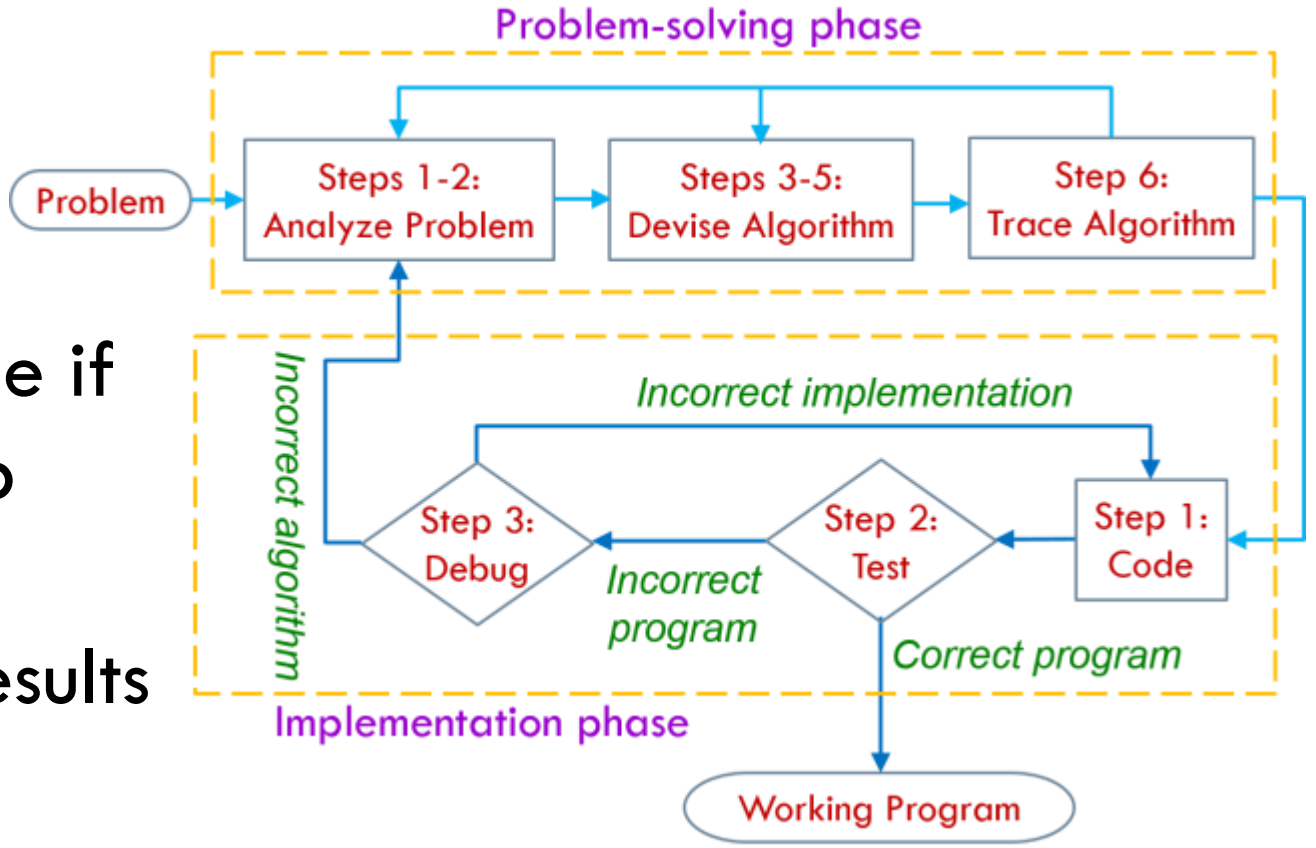
- Generalize steps into algorithm
  - Set $n = x, i = 1$
  - while $(i < y)$
    - $n = n * x$
    - $i = i + 1$
  - endwhile
  - n is answer

Algorithm for x^y (8/8)

- Test algorithm with broader variety of data

Implementation Phase

- Code the algorithm
- Test code
- Debug code if testing step generates incorrect results



Coding the Algorithm


```c
#include <stdio.h>

int main(void) {
  int x, y;
  printf("Enter two integers: ");
  scanf("%d %d", &x, &y);

  int n = x;
  int i = 1;
  while(i < y) {
    n = n * x;
    i += 1;
  }
  printf("%d raised to power of %d is: %d\n", x, y, n);
  return 0;
}
```

Coding the Algorithm – Even better


```c
#include <stdio.h>

int exponent(int x, int y);

int main(void) {
  printf("Enter base and power: ");
  int base, power;
  scanf("%d %d", &base, &power);
  int result = exponent(base, power);
  printf("%d ^ %d is: %d\n", base, power, result);
  return 0;
}

int exponent(int x, int y) {
  int n = x, i = 1;
  while (i < y) {
    n = n * x;
    i = i + 1;
  }
  return n;
}
```

Testing the Code

- Whitebox vs. Blackbox testing
- Unit test requires calling function or program and verifying that results are correct
- If testing step generates incorrect results, debug code
- Note: There are 2 major bugs in previous program!!!