

PRG1



NGEE ANN
SCHOOL OF INFOCOMM TECHNOLOGY

**W
E
E
K

1**

Problem Solving In Computing

Programming I (PRG1)

Diploma in Information Technology

Diploma in Financial Informatics

Diploma in Cybersecurity & Digital Forensics

Common ICT Programme

Year 1 (2019/20), Semester 1

Objectives

At the end of this lecture, you will be able to

- ☐ **apply steps to solve problems in computing**

What Is Problem Solving?

- People make decisions everyday to solve problems that affect their lives
 - ✓ Which polytechnic to go?
 - ✓ Which course to take?
 - ✓ Where to go for lunch after class?



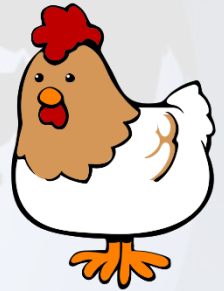
whatisengineering.com

Activity 1

Farmer Joe bought a fox, a chicken and a bag of beans. He hires a boat for \$20 to cross a river that is 50m wide, but can carry only one item at a time.

He needs to bring the purchases across the river to get home.

If left unguarded, the fox would eat the chicken, and the chicken would eat the beans.



How can you solve the problem that Farmer Joe faced – to get his purchases across the river?

Problem Solving Process

**Identify and Define
the problem**

**Analyze
the problem**

**Identify
possible solutions**

**Select and Plan
the solution**

**Implement & Evaluate
the solution**

- **5-step problem solving process that involves brainstorming, analyzing and testing until the optimal solution is reached**
- **It is important that all steps be done in order**

What Is A Problem?

- Setting out to achieve some objective or desired state of affairs and can include avoiding a situation or event
- All problems have two features in common:
 - Goals
 - Barriers



advanced-pcs.com

Setting Goals

- **Goals are:**
 - ✓ anything that you wish to achieve
 - ✓ where you want to be



Know Your Barriers

- **Barriers**

- ✓ Problem solving involves overcoming the barriers or obstacles that prevent the immediate achievement of goals



blog.pistolstar.us

Step 1. Identify And Define Problem

- **Recognize the existence of a problem**
 - ✓ Separate out the irrelevant and focus on the relevant issues that can solve the problem
- **Identify and name the problem**
 - ✓ Write down the problem in a clear and concise statement

Problem

Step 1. Identify And Define Problem

- **Set a goal**
 - ✓ Focus on what is the successful end of the process
 - ✓ Making a goal statement also helps to see the gap between the problem and the goal

Goal

Step 2. Analyze The Problem

- Analyze it to see what is involved in the problem
- Gather information
 - ✓ If there is not enough information, figure out how to research and collect it

Barriers

Step 3. Identify Possible Solutions

- **Develop possible solutions**
 - ✓ Brainstorm to identify all possible solutions or variations
- **Allow time for clarifications or suggestions**
 - ✓ Achieve common understanding

Step 4. Select And Plan The Solution

- **Select the best solution to fix the problem given the circumstances, resources, and other considerations**
 - ✓ Things that can effect a solution: money, time, people, procedures, policies, rules, and so on
- **Prioritize the solutions into what would work best**
 - ✓ This is a slow process of elimination
 - ✓ Discuss those solutions and come to consensus on the best solution

Step 5. Implement & Evaluate The Solution

- **Create an action list**
 - ✓ What needs to be done and by whom?
 - ✓ Step-by-step instructions that can be understood and executed
- **Evaluate the results to determine if it is the best possible solution to the problem**

The background of the slide features a light gray, semi-transparent image of a globe centered on the African continent. Overlaid on the left side of the globe is a faint, perspective-view image of a computer keyboard. The title 'Problem Solving in Computing' is written in a large, bold, purple font, centered over the globe.

Problem Solving in Computing

Problem Solving in Computing

Generic

**Identify and Define
the problem**

**Analyze
the problem**

**Identify
possible solutions**

**Select and Plan
the solution**

**Implement & Evaluate
the solution**

Computing

Understand the Problem

Formulate a Model

Develop an Algorithm

Write the Program

Test & Evaluate the Program

STEP 1: Understand The Problem

- You need to know:

- ✓ What input data/information is available?
- ✓ Do I have everything that I need?
- ✓ What format is it in?
- ✓ What output information am I trying to produce?
- ✓ What do I want the result to look like ... text, a picture, a graph ...?
- ✓ What am I going to have to compute?



STEP 1 - Example

Calculate the body mass index (bmi) of John and display the result on the screen.

✓ Input:

✓ Processing:

✓ Output:

STEP 2: Formulate A Model

- It is all about figuring out how to make use of the available data to compute an answer
- E.g., to compute the bmi of John based on input (height & weight)
 - ✓ Need to know the model (or formula) for computing bmi

$$\frac{\text{weight}}{\text{height} \times \text{height}}$$

- ✓ If there is no such "formula", we need to develop one

STEP 3: Develop An Algorithm

- Have a precise plan of what we want the computer to do using an **algorithm**
 - ✓ Informally, an algorithm is like a recipe
 - ✓ A set of steps that must be written in an unambiguous and precise manner
- An algorithm to solve a computer-based problem consists of 3 phases:
 - ✓ Input: what is available for the algorithm to solve
 - ✓ Process: how to solve the problem
 - ✓ Output: what is the required result



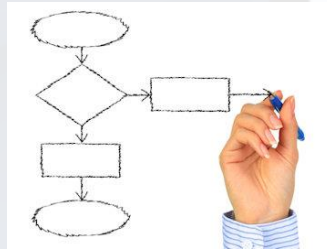
STEP 3: Develop An Algorithm

❑ 2 commonly used representations for an algorithm

✓ Pseudocode

- A simple and concise sequence of English-like instructions to solve a problem

✓ Flowcharts



STEP 3 - Example

Calculate the body mass index of John and display the result on the screen.

Pseudocode

1. Prompt user for the height in m
2. Get height
3. Prompt user for the weight in kg
4. Get weight
5. Calculate $bmi = \text{weight} / (\text{height} * \text{height})$
6. Display weight, height & bmi

STEP 4: Write The Program

- Transform the algorithm from Step 3 into a set of instructions that can be understood by the computer
- Writing a program is often called "*writing code*" or "*implementing an algorithm*"
 - ✓ The *code* (source code) is actually the program itself



STEP 4 - Example

Pseudocode

1. Prompt user for the height in m
2. Get height
3. Prompt user for the weight in kg
4. Get weight
5. Calculate $bmi = \text{weight} / (\text{height} * \text{height})$
6. Display weight, height & bmi

This program calculates the body mass index of a person

```
height = float(input('Enter your height in m: '))  
weight = float(input('Enter your weight in kg: '))
```

```
bmi = weight / (height * height)
```

```
print('Your height is ' + str(height) + 'm')  
print('Your weight is ' + str(weight) + 'kg')  
print('Your bmi is ' + str(bmi))
```

STEP 5: Test & Evaluate The Program

- **Ensure that program solves the problem that it was intended to solve and that the solutions are correct**
 - ✓ When you run (execute) your program, you should see the correct output

STEP 5: Example

```
# This program calculates the body mass index of a person
```

```
height = float(input('Enter your height in m: '))  
weight = float(input('Enter your weight in kg: '))
```

```
bmi = weight / (height * height)
```

```
print('Your height is ' + str(height) + 'm')  
print('Your weight is ' + str(weight) + 'kg')  
print('Your bmi is ' + str(bmi))
```

Output

```
>>> =====  
>>>  
Enter your height in m: 1.65  
Enter your weight in kg: 60  
Your height is 1.65m  
Your weight is 60.0kg  
Your bmi is 22.03856749311295  
>>> =====  
>>>  
Enter your height in m: 1.90  
Enter your weight in kg: 70  
Your height is 1.9m  
Your weight is 70.0kg  
Your bmi is 19.390581717451525  
>>>
```


STEP 5: Test & Evaluate The Program

- It is possible that your program works correctly for some set of data input but not for all
- If the output of your program is incorrect:
 - ✓ It is possible that algorithm was not converted properly into a proper program
 - ✓ It is also possible that a proper algorithm was not done e.g. Some instructions may be out of sequence

STEP 5: Test & Evaluate The Solution

- **Once your program produces a result that seems correct, you need to re-consider the original problem and make sure that the answer is formatted into a proper solution to the problem**
 - ✓ Program solution may not solve the problem the way that you wanted it to
 - ✓ More steps may be involved



dreamstime.com

Activity 1

State the input, processing and output needed to solve the following problems:

- Calculate the annual income for an executive employee.

✓ Input:

✓ Processing:

✓ Output:

Activity 2

Write the pseudocode for the problem:

- Calculate the annual income for an executive employee.**

Reading Reference

❑ How to Think Like a Computer Scientist: Learning with Python 3

✓ Chapter 1

http://openbookproject.net/thinkcs/python/english3e/way_of_the_program.html

❑ Free course What is Programming?

✓ Introduction

<https://www.pluralsight.com/courses/what-is-programming>

Summary

- **Problem Solving in Computing**
 - Uses programming language to communicate instructions to computers (also known as programming)
- **Problem Solving & Algorithms**
 - Algorithm defines sequence of concise & unambiguous steps or instructions to solve a problem