

Techniques for Solving Problems

In life we encounter problems every day, from deciding what to wear in the morning to planning a trip for summer vacation. Some problems may be easier to solve than others because we have a preference (I love to eat eggplant parmesan. So, where should we eat tonight?) while others may be very taxing (I have only one stick of bubble gum and two friends. Who will get it?).

This chapter will give you the tools to become a great problem solver. We will look at formalizing the steps you can take to systematically arrive at a solution, and how you can use these skills to program a computer.

I. Steps in solving a problem

While every problem is unique, there are some core steps that you can follow to help you break down and solve any situation. The steps that are outlined here will be used throughout the entire text for you to frame your programming solutions.

After you have read or listened to the problem presented,

1. Identify your **input**.

Input is the essential information that a program needs to solve the problem. The input may be given by an outside source (like the person using the program) or default values given by the programmer.

2. Identify the **goal** or **objective**.

The goal is where you define your end result. This is a description of when you know your problem has been solved and your objectives have been met. Sometimes you may have more than one goal, so list each goal clearly.

3. Create a list of **tasks** to achieve your objective.

These are the steps required to reach your goal. Make sure you enumerate your steps in order for clear execution. Also make sure each step is

descriptive and clear. Your list of tasks will result in your goal from step 2.

When you have finished your tasks, make sure that you have answered the problem and met your goal(s). This is probably the most important part of the steps, but the easiest one to forget. Remember if your steps (from part 3) don't lead to your goal (from part 2) then you didn't solve the problem and you should start back at step 1.

II. Motivating example

Juanita is planning a trip to visit the Grand Canyon. Since she lives in Seattle, Washington, she is concerned about the cost of gas needed to make the trip. Using online mapping software, she calculated that it will take 1,224 miles to reach the Grand Canyon. Traveling on the highway, Juanita's car gets 18 miles per gallon (*MPG*). Estimating that the average cost of gas is \$4.20 per gallon (*CPG*), calculate how much money it will cost for her to travel to the Grand Canyon. Using the equation below, you can calculate the cost of travel by distance divided by MPG times CPG:

$$\text{cost} = \text{distance} \div (\text{MPG} \times \text{CPG})$$

To solve this with our problem-solving steps from section 1.1, we first need to identify the input (step 1):

distance: 1224 [miles]

MPG: 18

CPG: \$4.20

Now we can state our goal (step 2):

Find the cost

Lastly, we list our tasks to reach our goal (step 3):

1. Plug the input(s) into the cost equation
2. Return the cost

Now that the problem is broken down into some clear steps, we can execute the tasks to reach our goal.

$$\text{cost} = 1224 / 18 * 4.20$$

$$\text{cost} = \$285.60$$

III. A Complete Problem

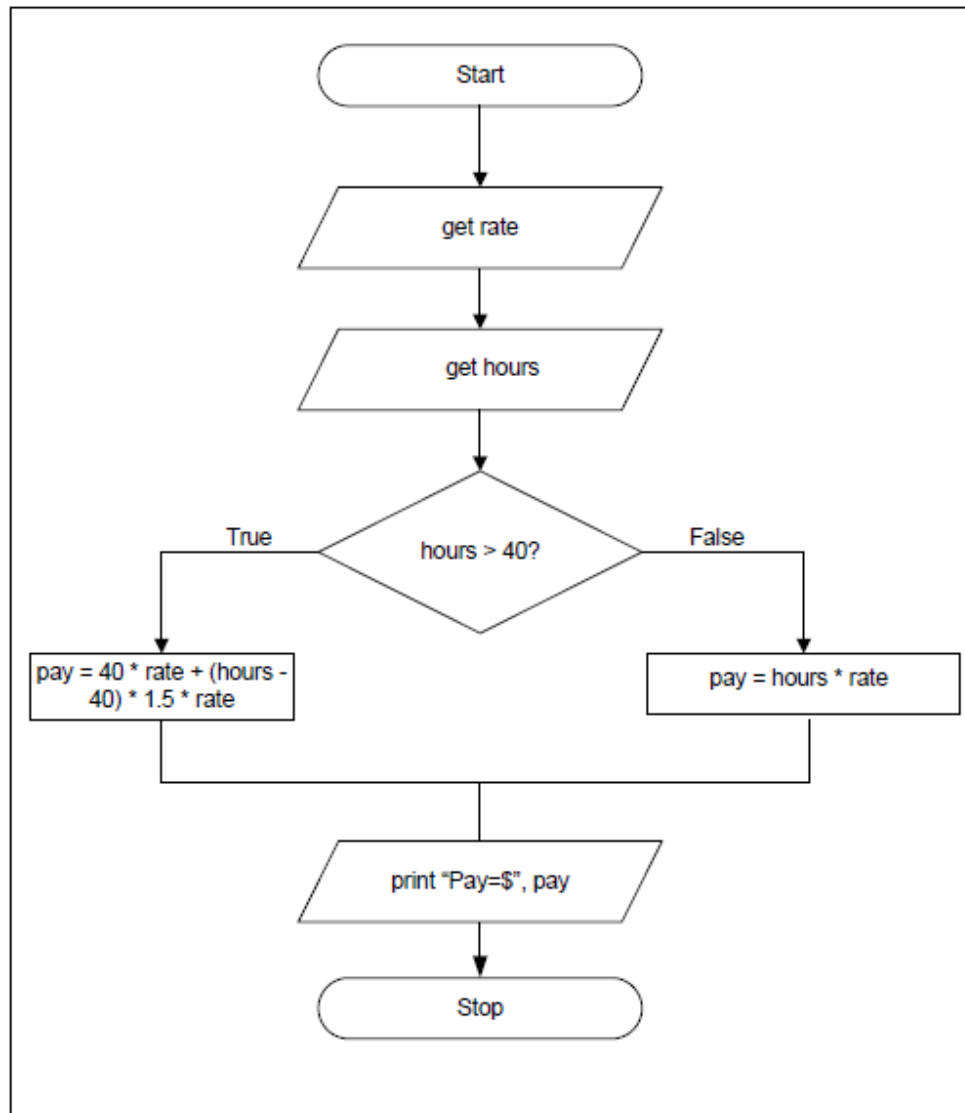
We need to read in the rate of pay (in dollars per hour) and the number of hours an employee has worked for a week. Calculate the amount the employee should be paid according to the following rules:

- (a) Regular pay: up to 40 hours, at the given rate.
- (b) Overtime pay: for each hour above 40, at 1.5 times the given rate.

Step by step:

- 1) Identify your input:
 - Rate of pay.*
 - Number of hours worked a week.*
- 2) Identify the **goal** or **objective**.
 - Amount the employee should be paid.*
- 3) Create a list of **tasks** to achieve your objective.
 - Get the rate of pay from the user.
 - Check if the number of hours is greater than.
 - If the number of hours is greater than 40, then apply the formula:
 - $\text{Pay} = 40 \times \text{rate} + (\text{hours} - 40) \times 1.5 \times \text{rate}$
 - Show this amount to the user.
 - If the number of hours is less or equal than 40, then apply the formula:
 - $\text{Pay} = \text{hours} \times \text{rate}$
 - Show this amount to the user.
 - End of instructions

The flowchart looks like this:



And the pseudocode is as follows:

```
Start
  print "Please enter rate of pay: "
  get rate
  print "Please enter hours worked: "
  get hours
  If hours > 40 Then
    pay = 40 * rate + (hours - 40) * 1.5 * rate
  Else
    pay = hours * rate
  EndIf
  print "Pay=$" + pay
Stop
```

Finally, the Python program is this:

```
#This program calculates and prints pay given
#rate and hours worked
rate = float(input("Enter rate: "))
hours = float(input("Enter hours worked: "))
if hours > 40:
    pay = 40 * rate + (hours - 40) * 1.5 * rate
else:
    pay = hours * rate
print ("Pay=$", pay)
```

IV. Topics you may want to choose

1. Alice Springs is a remote town in Australia's Northern Territory, supporting a population of over 25,186 people (in the 2011 census). The large population and desert location make water a treasured resource. Many citizens of Alice Springs do not own their own swimming pool because water is so dear; instead they visit the public swimming pools. If the city charges 1.2 cents per cubic meter of water, calculate the cost the city pays for water needed to fill a typical public swim pool if the pool is 10 meters wide, 20 meters long and 6 meters deep. Use the following equations:

$$\text{Volume in cubic meter} = \text{length} * \text{width} * \text{height}$$
$$\text{Cost} = \text{cost per cubic meter} * \text{Volume in cubic meter}$$

Your code should support a menu to enter the dimensions of the pool, and a list containing few cities and their corresponding charges per cubic meter of water.

2. You have decided to repaint the four walls of your living room and need to know how many gallons of paint to buy. There is one window and one entrance to the room. Describe how you would calculate the number of gallons of paint needed taking the description of the room into account. Assume one gallon of paint covers 100 square meters.

Your code should support a menu to enter the capacity of the gallon of paint, and a list containing some living-spaces (bathroom, kitchen, living room) alongside their corresponding dimensions.

3. You have decided to enter a model boat race. You put your boat at the start line next to your best friend Jill's boat. Create a program to print out the distance that both of your boats travelled given the speed your boat travels in (meter per minute), the speed that Jill's boat travels and the number of minutes in the race.

Your code should support a menu to enter the speed of the two boats, and a list containing different race circuits, including their length.