Chapter 3

Dr. Raza Ul Mustafa Khokhar
American University
Computer Science Department - CSC-148

How to be a Successful Programmer

- As programmers we spend 99% of our time trying to get our program to work. We struggle, we stress, we spend hours deep in frustration trying to get our program to execute correctly
- Essentially I'm telling you once again, start small, get something small working, and then add to it.

How to Avoid Debugging

- Understand the Problem
  - You must have a firm grasp on what you are trying to accomplish but not necessarily how to do it.
- Start Small
  - Start with something really small. Maybe just two lines and then make sure that runs.
- Keep Improving it
  - Once you have a small part of your program working, the next step is to figure out something small to add to it – how can you move closer to a correct solution.

Let’s look at an example

- Ask the user for the time now (in hours 0 - 23), and ask for the number of hours to wait.
- ***Your program should output what the time will be on the clock when the alarm goes off.***
- For example, if current_time is 8 and wait_time is 5, final_time should be 13 (1 pm).

Solution next slide →

```
current_time = input("what is the current time (in hours)?")
wait_time = input("How many hours do you want to wait")

print(current_time)
print(wait_time)
```

Solution

- Is there any error here?

```
current_time = input("What is the current time (in hours 0 - 23)?")
wait_time = input("How many hours do you want to wait")

print(current_time)
print(wait_time)

final_time = current_time + wait_time
print(final_time)
```

One solution ?

- So what do we do about the problem? We will need to convert both current_time and wait_time to int.

```
current_time_str = input("What is the current time (in hours 0-23)?")
wait_time_str = input("How many hours do you want to wait")

current_time_int = int(current_time_str)
wait_time_int = int(wait_time_str)

final_time_int = current_time_int + wait_time_int
print(final_time_int)
```

Or maybe another solution?

- Now, that's a lot better, and in fact depending on the hours you chose, it may be exactly right. If you entered 8 for current_time and 5 for wait_time then 13 is correct.

- **But if you entered 17 (5 pm) for current_time and 9 for wait_time then the result of 26 is not correct.**

Solution?

Solution

```
current_time_str = input("What is the current time (in hours 0-23)?")
wait_time_str = input("How many hours do you want to wait")

current_time_int = int(current_time_str)
wait_time_int = int(wait_time_str)

final_time_int = current_time_int + wait_time_int

final_answer = final_time_int % 24

print("The time after waiting is: ", final_answer)
```

Beginning tips for Debugging

- The process of debugging is much more like being a detective

Error Messages

Print Statements

Know Your Error Messages, What is error?

```
current_time_str = input("What is the current time (in hours 0-23)?")
wait_time_str = input("How many hours do you want to wait")

current_time_int = int(current_time_str)
wait_time_int = int(wait_time_int)

final_time_int = current_time_int + wait_time_int
print(final_time_int)
```

Errors

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_21032\1154067393.py in <module>
      3
      4 current_time_int = int(current_time_str)
----> 5 wait_time_int = int(wait_time_int)
      6
      7 final_time_int = current_time_int + wait_time_int

NameError: name 'wait_time_int' is not defined
```

https://shorturl.at/doqHS : Different types error statistics

Parse error

- Parse errors happen when you make an error in the syntax of your program.

```
current_time_str = input("What is the current time (in hours 0-23)?")
wait_time_str = input("How many hours do you want to wait"

current_time_int = int(current_time_str)
wait_time_int = int(wait_time_str)

final_time_int = current_time_int + wait_time_int
print(final_time_int)
```

Type error

- TypeErrors occur when you you try to combine two objects that are not compatible. For example you try to add together an integer and a string.

```
a = input('wpisz godzine')
x = input('wpisz liczbe godzin')
int(x)
int(a)
```

Name errors

- Name errors almost always mean that you have used a variable before it has a value.
- Often NameErrors are simply caused by typos in your code.

```
str_time = input("What time is it now?")
str_wait_time = input("What is the number of nours to wait?")
time = int(str_time)
wai_time = int(str_wait_time)

time_when_alarm_go_off = time + wait_time
print(time_when_alarm_go_off)
```

Another name error example

```
n = input("What time is it now (in hours)?")
n = imt(n)
m = input("How many hours do you want to wait?")
m = int(m)
q = m % 12
print("The time is now", q)
```

Value errors

- Value errors occur when you pass a parameter to a function and the function is expecting a certain limitations on the values, and the value passed is not compatible

```
current_time_str = input("What is the current time (in hours 0-23)?")
current_time_int = int(current_time_str)

wait_time_str = input("How many hours do you want to wait")
wait_time_int = int(wait_time_int)

final_time_int = current_time_int + wait_time_int
print(final_time_int)
```