



Programming

Inputs, Outputs, Variables and Data Types

GCSE Student Booster

Key Information

- 1) Remember this booster is here to **help you**. Please consider your behaviour in the chat.
- 2) If you are in a room with a teacher/group, please login to the meeting. This is so we can mark your attendance. This information goes into a **prize draw**.
- 3) Make sure the name on the meeting is the **SAME** as the name on your Isaac account. We can't mark you present if they don't match.

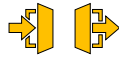


Learning Outcomes

- Understand how to use outputs in programming
- Understand how to use inputs in programming
- Understand the different data types and when to use each data type
- Understand what is meant by a variable and how to use them in programming
- Understand how to use string concatenation and casting



Programming – Input, Output, Data Types Isaac Booster



DO NOW

Rearrange the code below to make a working program. Write the correct order of line numbers in the chat.

```
1 print("You guessed " + str(guess) + " and the answer was " + str(answer))
2 answer = 42
3 guess = int(input)
4 print("Guess the secret number, it's between 1 and 100.")
```



Programming – Input, Output, Data Types Isaac Booster



DO NOW SOLUTION

```
print("Guess the secret number, it's between 1 and 100.")
guess = int(input)
answer = 42
print("You guessed " + str(guess) + " and the answer was " + str(answer))
```

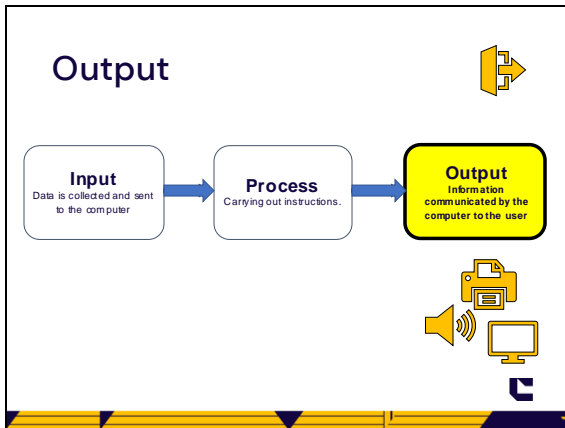


Output



Information communicated by
the computer to the user





Output

Information communicated to the user by the computer.

Python	<code>print("Hello world")</code>
OCR ERL	<code>print("Hello world")</code>
AQA Pseudocode	OUTPUT "Hello world"

print() – breaking it down

The **statement**. Like a command word for the computer.
This part of the code tells it what task to perform.

`print("your text goes here")`

print() – breaking it down

The **argument**. The data that the print statement works with.

This example uses a **string literal**, which is the programming term for text enclosed in quotation (speech) marks.

```
print("your text goes here")
```

Activity – Predict the Output

1. `print("I love Computing")`

```
I love Computing
```

2. `print("I love Comping")`

```
I love Comping
```

3. `print("I love Computing"`

```
print("I love Computing"
^
SyntaxError: '(' was never closed
```

Activity – Debugging

```
Print "I love Computing)
```

The code example above contains several syntax errors. Open the link below and find and correct the errors.

[Code link](#)

Debugging – Solution

Print "I love Computing)

```
print("I love Computing")
```

Did you get all three errors?



Concatenating Strings

We can use the + character to join strings together before printing, like this:

```
>>> print("hello" + "world")
helloworld
>>> print("3" + "2" + "1")
321
```

This is called **concatenation**. Note there is no space between strings in the output!



Outputting multiple items

The Python print() function can also take many **arguments** separated by commas, like this:

```
>>> print("hello", "world")
hello world
>>> print("3", 2, 1)
3 2 1
```

This time, Python assumes you want a **space** between the values outputted. Also, you can mix strings and other types of data.



Variables

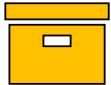


A named space in memory used to store one piece of data whilst the program is running.



Programming - Variables

favColour



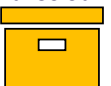
← "blue"



Variable Assignment

Storing data in a variable

favColour



← "blue"

```
favColour = "blue"  
favColour ← "blue"
```



Variable Identifier



The name you have chosen for the variable

```
favColour = "blue"
```

The **identifier** of the variable where we wish to store the data. This goes on the LEFT.



=



*The operator for **variable assignment**.*

```
favColour = "blue"  
favColour ← "blue"
```

Goes in the MIDDLE. Whenever you see this symbol, variable assignment is being used.



Data



The information being stored in the variable

```
favColour = "blue"
```

Goes on the RIGHT. Whatever is here will be stored in the variable named on the left.



Activity – spot the assignments

How many times is assignment used in this code? Type a number in the **chat** now.

```
1 correct_password = "LetMeIn"
2 user_password = "123456"
3
4 if user_password == correct_password:
5     print("Access granted")
6 else:
7     print("Access denied")
```

Expressions

We can assign the result of an **expression** to a variable.

These are all valid assignment statements including expressions:

```
total = 10 + 6
average = total / 2
greeting = "Hello " + name
```

Variable – Key Information

A named space in memory used to store one piece of data, whilst the program is running, and its value can change.

- Its name can be anything but should be **meaningful**, e.g. **favColour** not **fc**, and it can't contain spaces.
- Is given a value by an **assignment** statement. This can be a **literal** or the result of an **expression**.
- Can change, but only stores one piece of data at a time, so each assignment **overwrites** the contents.
- We can't have 2 variables with the same identifier.

Outputting a variable

We can now use **variables** in an output statement, as well as literals.

```
favColour = "blue"
print("fave colour is " + favColour)

age = 14
print("You are", age, "years old!")

newage = age + 1
print("Next year you will be ", newage)
```

Activity – trace assignments

What data is in the variables **num1**, **num2** and **num3** after the whole program has run? Type in the **chat**.

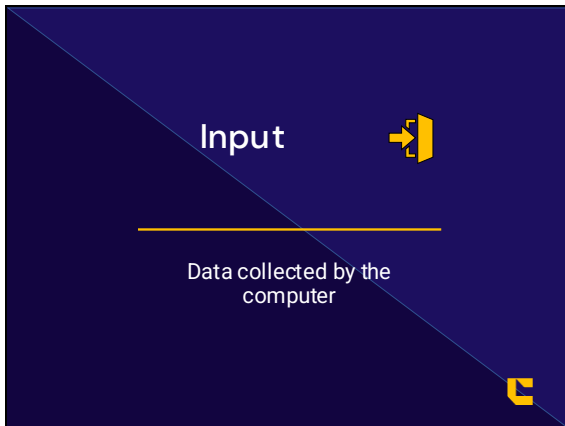
```
1 num1 = 42
2 num2 = 99
3 num2 = 35
4 num1 = num2
5 num3 = 3
6 num2 = 1337
```

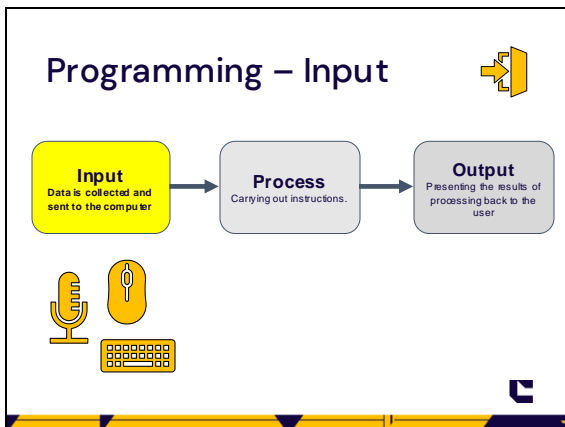
Activity – what's the output?

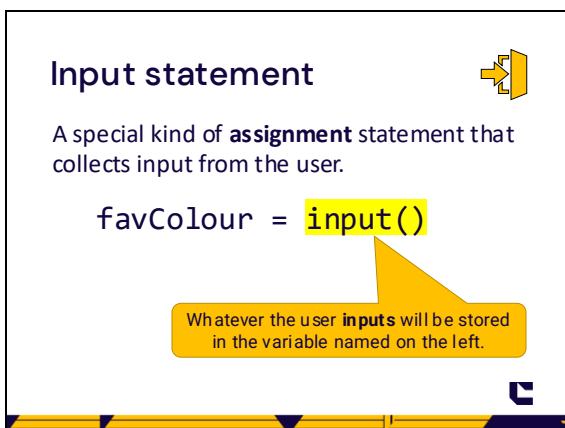
What will be output by this program?

```
computer_name = "Hal"
user_name = "Adil"
print("Hello " + user_name)
print("My name is", computer_name)
```

```
Shell x
>>> %Run output_test.py
Hello Adil
My name is Hal
```







Using input, variable and output together

1. Output the question.

```
1 print("What's your name?")
```



Using input, variable and output together

1. Output the question.
2. Assign **input** to a variable.

```
1 print("What's your name?")
2 name = input()
```



Using input, variable and output together

1. Output the question.
2. Assign input to a variable.
3. Use the variable in an **output** statement if needed.

```
1 print("What's your name?")
2 name = input()
3 print("Hello " + name)
```



PRIMM: Predict, Run, Investigate, Modify, Make



PRIMM Worked Example – Predict



```
1 print("What's your favourite colour?")
2 fav_colour = input()
3 print("I love " + fav_colour + " too!")
```

What will this program output if the user inputs **purple**? Write a prediction:

*It will output 'I love **purple** too'.*

[Code link](#)



PRIMM Worked Example – Run



Run and test my prediction:

```
What's your favourite colour?
purple
I love purple too!
```

My prediction was mostly correct, but I missed the exclamation mark from the end of the output.



PRIMM Worked Example – Investigate



```
1 print("What's your favourite colour?")
2 fav_colour = input()
3 print("I love " + fav_colour + " too!")
```

What **variable** is used in this code?

Where is **assignment** used?

*fav_colour is the only variable
Line 2 is where assignment is used*



PRIMM Worked Example – Modify



```
1 print("What's your favourite colour?")
2 fav_colour = input()
3 print("I love " + fav_colour + " too!")
```

Change the code to ask what their favourite **sport** is instead of colour.

```
1 print("What's your favourite sport?")
2 fav_sport = input()
3 print("I love " + fav_sport + " too!")
```



PRIMM Worked Example – Make



```
1 print("What's your favourite colour?")
2 fav_colour = input()
3 print("I love " + fav_colour + " too!")
```

Make a new program that asks the user two questions, their favourite food and favourite day of the week, and outputs an appropriate response.

```
print("What's your favourite food?")
fav_food = input()
print("What's your favourite day?")
fav_day = input()
print("I always eat " + fav_food +
      " on " + fav_day + "!")
```



PRIMM Worked Example – Make



```
1 print("What's your favourite colour?")
2 fav_colour = input()
3 print("I love " + fav_colour + " too!")
```

Make a new program that asks the user two questions, their favourite food and favourite day of the week, and outputs an appropriate response.



Activity – PRIMM



Handout 1

Open Handout 1 - Input Task Sheet

Complete the PRIMM tasks

We will stop and check answers after each task.

Programming – Output, Input, Variables, Data Types

Handout 1 - Input Task sheet

Task 1 - Predict Task

```
1 print("What's your favourite subject?")
2 fav_subject = input()
3 print("What's your favourite music?")
4 fav_music = input()
5 print("What's your favourite chocolate?")
6 fav_chocolate = input()
7 print("This person likes " + fav_subject + " and " + fav_chocolate)
```

My Prediction

Once you have written your prediction, [click here to open the program](#) and take a screenshot of it running.

If there are any differences from your prediction then explain them in the final column.

Screenshot of the code running	Differences from my prediction



Task 1 – Predict

Handout 1
90 seconds

Predict what the code below will output based on the inputs provided.


```
1 print("What's your favourite subject?")
2 fav_subject = input()
3
4 print("What's your favourite music?")
5 fav_music = input()
6
7 print("What's your favourite chocolate?")
8 fav_chocolate = input()
9
10 print("This person likes " + fav_subject + " and " + fav_chocolate)
```



Task 1 – Predict – Solution

Handout 1

```
1 print("What's your favourite subject?")
2 fav_subject = input()
3
4 print("What's your favourite music?")
5 fav_music = input()
6
7 print("What's your favourite chocolate?")
8 fav_chocolate = input()
9
10 print("This person likes " + fav_subject + " and " + fav_chocolate)
```

Powered by  **trinket**
 What's your favourite subject?
 drama
 What's your favourite music?
 indie
 What's your favourite chocolate?
 maltesers
 This person likes drama and maltesers



Task 2 – Run

Handout 1

1 minute

Now you will run the code and check it against your prediction.

Complete the table provided and insert a screenshot.

Screenshot of the code running	Differences from my prediction.



Task 3 – Investigate

Handout 1

3 minutes

Answer the questions based on the code screenshot provided.

```
1 print("What's your favourite subject?")
2 fav_subject = input()
3
4 print("What's your favourite music?")
5 fav_music = input()
6
7 print("What's your favourite chocolate?")
8 fav_chocolate = input()
9
10 print("This person likes " + fav_subject + " and " + fav_chocolate)
```



Task 3 – Investigate – Solutions



```
1 print("What's your favourite subject?")
2 fav_subject = input()
3
4 print("What's your favourite music?")
5 fav_music = input()
6
7 print("What's your favourite chocolate?")
8 fav_chocolate = input()
9
10 print("This person likes " + fav_subject + " and " + fav_chocolate)
```

1. What are the identifiers for the variables?
fav_subject, fav_music & fav_chocolate
2. What command is used to create output?
Print
3. How many times is variable assignment used in the code?
3



Task 3 – Investigate – Solutions



```
1 print("What's your favourite subject?")
2 fav_subject = input()
3
4 print("What's your favourite music?")
5 fav_music = input()
6
7 print("What's your favourite chocolate?")
8 fav_chocolate = input()
9
10 print("This person likes " + fav_subject + " and " + fav_chocolate)
```

4. Give the line numbers where input is used
2,5,8
5. Give the line number where concatenation is used.
10
6. What operator is used for concatenation?
+



Task 4 – Modify



2 minutes

Click the link and adapt the starter code below using the instructions provided.

```
1 name = "Billy"
2 print("We want to know if you like programming!")
3 print("Do you like programming " + name + "?")
4 answer = input()
5 print("Great! You said " + answer + "!")
6 print("Let's learn some Python today")
```





Task 4 – Modify – Solution

```
1 name = "Billy"
2 print("We want to know if you like programming!")
3 print("Do you like programming " + name + "?")
4 answer = input()
5 print("Great! You said " + answer + "!")
6 print("Let's learn some Python today")
```

```
1 print("What's your name?")
2 name = input()
3 print("We want to know if you like programming!")
4 print("Do you like programming " + name + "?")
5 answer = input()
6 print("Great, " + name + ". You said " + answer + "!")
7 print("Let's learn some Python today")
```

[Code link](#)





Task 5 – Make

3 minutes

Open the starter file and create your own program using the instructions provided.

Hint: You can use your answers to previous tasks to guide you as you write your code!

Powered by trinket

```
What's your name?
Bernard
What's your enemy's name?
Eunice
What's your superpower?
Invisibility
Where do you live?
Manchester
Hello Bernard!
Your power of Invisibility will mean that you're certain to defeat Eunice for good this time.
Walk tall down the streets of Manchester and use Invisibility for good, not evil!
```





Task 5 – Make – Solution

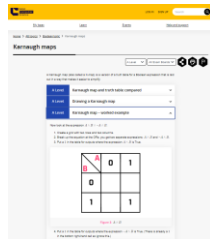
```
1 print("What's your name?")
2 name = input()
3
4 print("What's your enemy's name?")
5 enemy = input()
6
7 print("What's your superpower?")
8 power = input()
9
10 print("Where do you live?")
11 place = input()
12
13 print("Hello " + name + "!")
14 print("Your power of " + power +
15       " will mean that you're certain to defeat " + enemy +
16       " for good this time.")
17 print("Walk tall down the streets of " + place +
18       " and use " + power + " for good, not evil!")
19
```

[Code link](#)

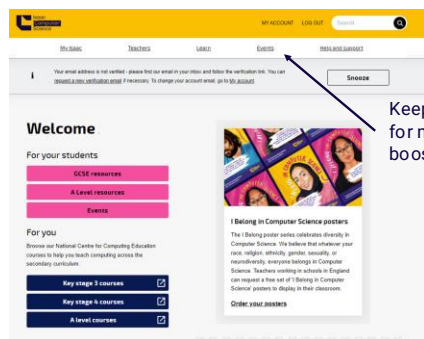


Isaac Computer Science

Don't forget,
IsaacComputerScience.org
 is your totally free online
 textbook and revision aid.
 It contains the entire GCSE
 and A-level syllabus and
 hundreds of high-quality
 self-assessment questions.



Check for more ISAAC boosters



Keep an eye out
 for more student
 booster events



Data Types

1010
1010

The type of data being stored or
 manipulated within a program



Data types



All data in a program has a **data type**. Data types are important because they determine what a program may do with the data.

For example, you can add two **numbers** together, but you can't add numbers to a word.

```
favColour = "blue"  
age = 14
```

```
newage = age + 1
```

```
newColour = favColour + 1
```



Common Data Types

Data type	Example
Integer	-5, 123, 0
Real (float)	1.1, -1.0, 382.0, 12.125
Boolean	True, False
Char	c, A, X, £
String	"Hello", "1\$34A", "ninety-four"



Task 1 – Choosing Data Types



What data types would be used to store the values in the table?

Value
Steve
15
007
B901LK
A
20.2
True
1



Task 1 – Choosing Data Types – Solution

Handout 2

Value	Data Type
Steve	String
15	Integer
007	String
B901LK	String
A	Char / String
20.2	Real / Float
True	Boolean
1	Integer

Casting



Converting data from one type
to another.

Inputting numbers

input() ALWAYS collects data as a **string**.
So we can't do this:

```
1 print("How old are you?")
2 age = input()
3 newage = age + 1
4 print("Next year you will be", newage)
```

```
Shell
How old are you?
15
Traceback (most recent call last):
  File "C:\Users\Public\test2.py", line 3, in <module>
    newage = age + 1
TypeError: can only concatenate str (not "int") to str
```

Casting input to integer

We need to change the data type of the input. We call this **casting**. We use `int()` to cast a string to an integer:

```
1 print("How old are you?")
2 age = int(input())
3 newage = age + 1
4 print("Next year you will be", newage)
```

```
How old are you?
15
Next year you will be 16
```



More casting functions

In Python there are many casting functions:

Data Type	Casting Function	Example	Result
Integer	<code>int()</code>	<code>int("15")</code>	15
String	<code>str()</code>	<code>str(2026)</code>	"2026"
Float	<code>float()</code>	<code>float("1.3")</code>	1.3



Task 2 – Fix The Casting Handout 2

4 minutes

Open the starter code and use the instructions to fix the program.

```
1 ## Adder Program ##
2
3 print("Enter a whole number")
4 num1 = input()
5
6 print("Enter a decimal number")
7 num2 = input()
8
9 total = num1 + num2
10
11 print("The total is " + total)
```

[Code link](#)



Task 2 – Fix The Casting – Solution

```

1  ## Adder Program ##
2
3  print("Enter a whole number")
4  num1 = int(input())
5
6  print("Enter a decimal number")
7  num2 = float(input())
8
9  total = num1 + num2
10
11 print("The total is " + str(total))

```

[Code link](#)

Task 3 – Make – Big Tipper

Some friends are going for dinner at a restaurant and want to add a tip to the bill then share the cost evenly. They haven't decided the tip yet, that depends on the service!



Handout 2
10 minutes

Task 3 – Big Tipper

Write a program that:

1. Gets inputs for the total cost of the meal in pounds and pence, (e.g. 85.40), tip percentage (e.g. 10 or 20) and number of persons eating (e.g.
2. Calculates the tip amount e.g. 10% or 20% and adds this percentage of the meal price to the total cost.
3. Calculates the cost per person.
4. Outputs the cost per person as part of a sentence.

```

Input cost of meal
65
Input tip percentage
10
Input number of diners
2
You will each pay £35.75

```



Task 3 Big Tipper Solution

```
# Gets inputs for the total cost of the meal, tip percentage and number of diners.
print("Input cost of meal")
cost = float(input())

print("Input tip percentage")
tip_percent = int(input())

print("Input number of diners")
num_of_diners = int(input())

# Calculates the tip amount and adds it to the total cost.
tip_amount = cost * (tip_percent/100)
total_cost = cost + tip_amount

# Calculates the cost per person.
cost_each = total_cost / num_of_diners

# Outputs the cost per diner as part of a sentence.
print("You will each pay £"+ str(cost_each))
```

Learning Outcomes

- Understand how to use outputs in programming
- Understand how to use inputs in programming
- Understand the different data types and when to use each data type
- Understand what is meant by a variable and how to use them in programming
- Understand how to use string concatenation and casting

Isaac Gameboard

Complete the gameboard at
bit.ly/iginputs

You will need to sign in to **Isaac Computer Science** or register for a free account if not done already.

