

**GCSE  
OCR**

Computer Science  
J277

5

**The Integrated  
Development  
Environment**

Unit 8  
Logic and languages



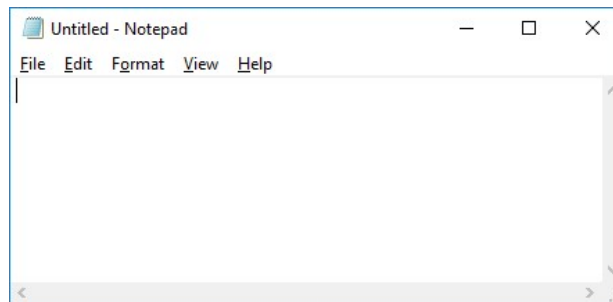
**PG ONLINE**

# Objectives

- Understand the use of an Integrated Development Environment (IDE) to develop programs, including the following:
  - Editors
  - Error diagnostics
  - Run-time environment

# Starter

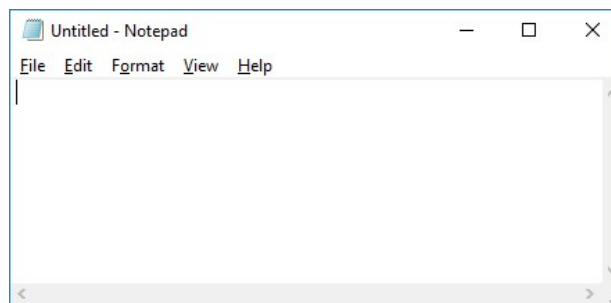
- The simplest way to edit the source code to programs is to use a text editor such as Notepad
  - What problems are there with using such a simple text editor?



# Starter

Answers

- There are many problems that such a simple text editor has, including:
  - No line numbers
  - No syntax highlighting (colour coding of different parts of code)
  - No ability to run code directly inside the editor
  - No assistance in debugging code



# IDEs

- IDEs are Integrated Development Environments
  - They have a number of tools and features that help programmers when they are programming
- Commonly used IDEs include Visual Studio (for C#, VB and other languages), Eclipse (for C++ and many more) and IDLE (for Python)
  - What features do IDEs provide?

# IDEs

- The following screenshot shows some of the features of an IDE
  - Explain what each of these features mean

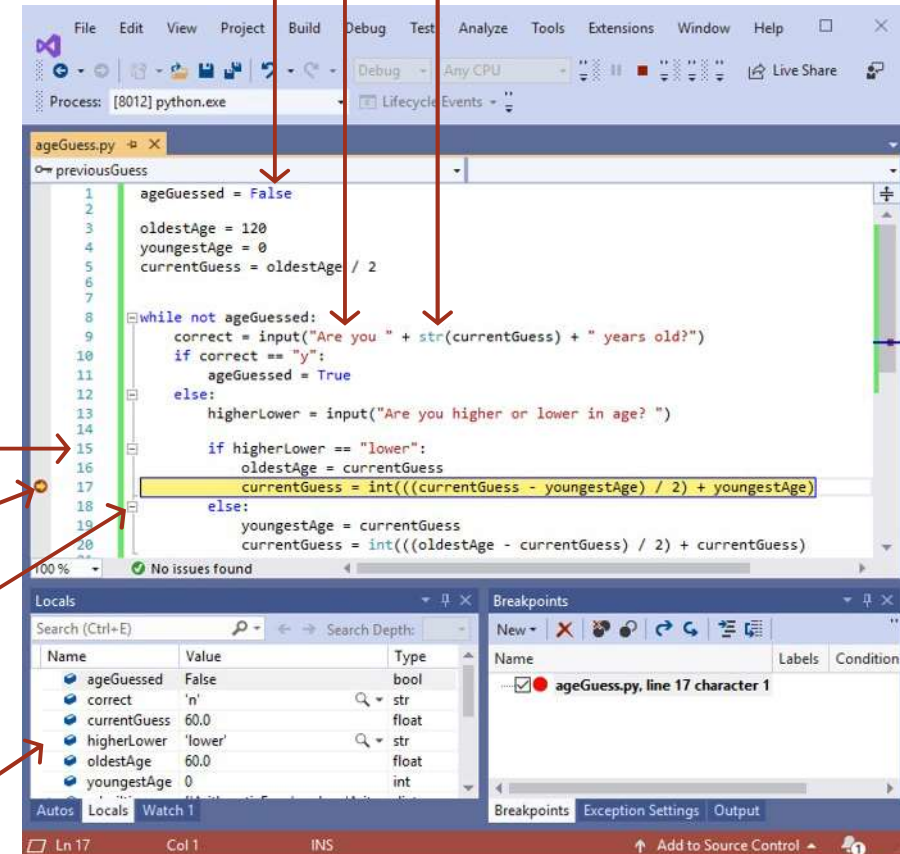
## Syntax highlighting

Line numbers

Breakpoints

Code folding

Variable watching



# Line numbers

- Line numbers allow a programmer to clearly see each new line of code
  - When errors are found, the line number that they occur on will also be stated
  - In some IDEs, parts of the program that the programmer doesn't need to see can be folded

```
1 ageGuessed = False
2
3 oldestAge = 120
4 youngestAge = 0
5 currentGuess = oldestAge / 2
6
7
8 while not ageGuessed:
9     correct = input("Are you " + str(currentGuess)
10    if correct == "y":
11        ageGuessed = True
12    else:
13        higherLower = input("Are you higher or lower
14
15        if higherLower == "lower":
16            oldestAge = currentGuess
17            currentGuess = int(((currentGuess - you
18        else:
19            youngestAge = currentGuess
```

```
1 ageGuessed = False
2
3 oldestAge = 120
4 youngestAge = 0
5 currentGuess = oldestAge / 2
6
7
8 while not ageGuessed:...
```

Notice the line numbers change where the code has been folded

# Syntax highlighting

- Syntax highlighting is where the colour of the text changes to show different parts of the program
  - What do each of the colours mean below?

```
1  ageGuessed = False
2
3  oldestAge = 120
4  youngestAge = 0
5  currentGuess = oldestAge / 2
6
7
8  while not ageGuessed:
9      correct = input("Are you " + str(currentGuess) + " years old?")
10     if correct == "y":
11         ageGuessed = True
12     else:
13         higherLower = input("Are you higher or lower in age? ")
14
15         if higherLower == "lower":
16             oldestAge = currentGuess
17             currentGuess = int(((currentGuess - youngestAge) / 2) + youngestAge)
18         else:
19             youngestAge = currentGuess
20             currentGuess = int(((oldestAge - currentGuess) / 2) + currentGuess)
21
22
```





# Syntax highlighting

Answers

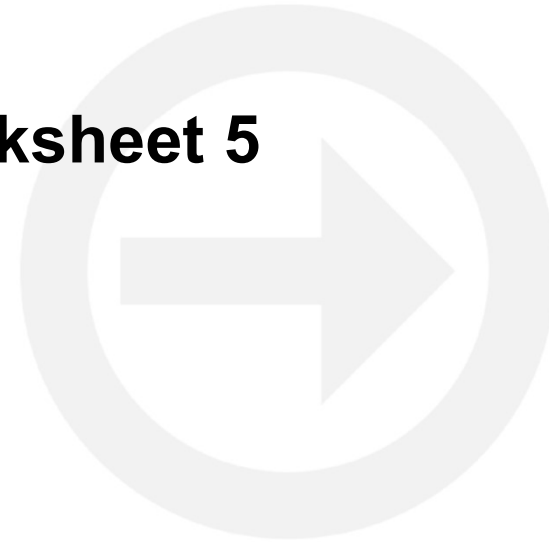
	Booleans (false, true), Keywords (while, not, if, else)
	Type conversions (str, int)
	Strings – e.g. "Are you higher or lower in age? "
	All other operators, variables and functions names

```
1  ageGuessed = False
2
3  oldestAge = 120
4  youngestAge = 0
5  currentGuess = oldestAge / 2
6
7
8  while not ageGuessed:
9      correct = input("Are you " + str(currentGuess) + " years old?")
10     if correct == "y":
11         ageGuessed = True
12     else:
13         higherLower = input("Are you higher or lower in age? ")
14
15         if higherLower == "lower":
16             oldestAge = currentGuess
17             currentGuess = int(((currentGuess - youngestAge) / 2) + youngestAge)
18         else:
19             youngestAge = currentGuess
20             currentGuess = int(((oldestAge - currentGuess) / 2) + currentGuess)
21
22
```



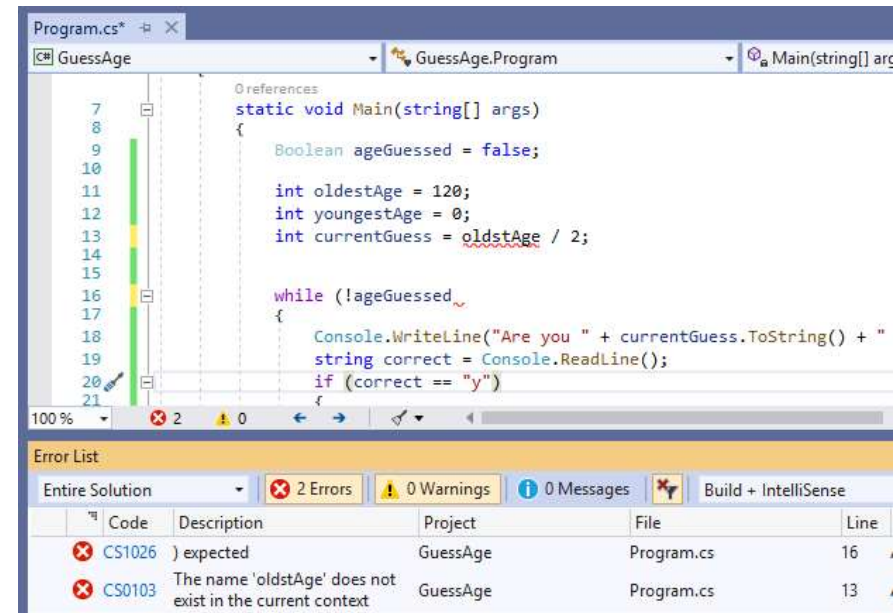
# Worksheet 5

- Now complete **Task 1** on **Worksheet 5**



# Error diagnostics

- Error diagnostics help a programmer to find where they have made a mistake
  - Errors are identified along with the line number that they occur on
  - The code may be underlined or highlighted to show the error
- This program is written C#
  - With the help of the error diagnostics fix the two errors shown



```
Program.cs* x
GuessAge GuessAge.Program Main(string[] args)
References
static void Main(string[] args)
{
    Boolean ageGussed = false;

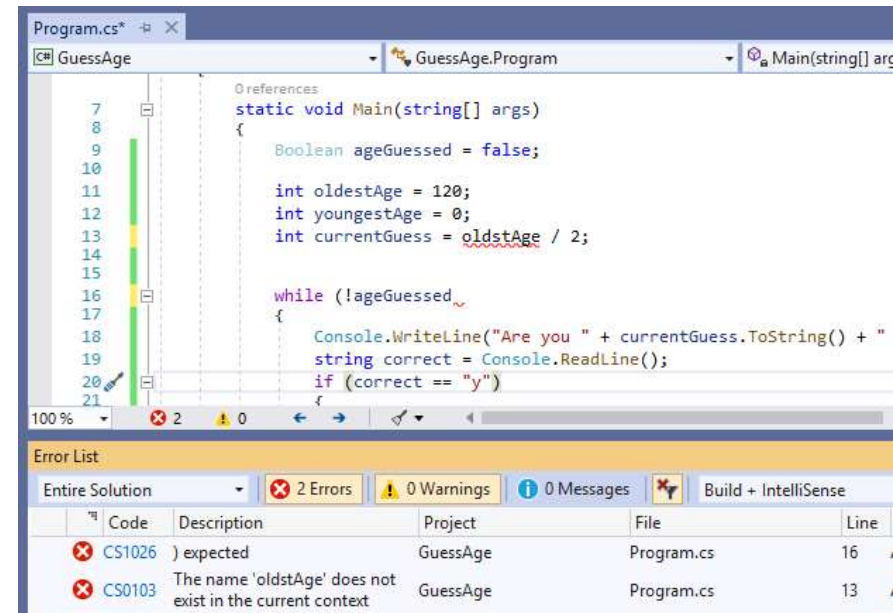
    int oldestAge = 120;
    int youngestAge = 0;
    int currentGuess = oldstAge / 2;

    while (!ageGussed
    {
        Console.WriteLine("Are you " + currentGuess.ToString() + "
        string correct = Console.ReadLine();
        if (correct == "y")
    }
}
100% 2 0
Error List
Entire Solution 2 Errors 0 Warnings 0 Messages Build + IntelliSense
Code Description Project File Line
CS1026 ) expected GuessAge Program.cs 16
CS1013 The name 'oldstAge' does not exist in the current context GuessAge Program.cs 13
```

# Error diagnostics

Answers

- The errors show that:
  - A bracket needs to be added to the end of line 16
  - The variable `oldstAge` in line 13 doesn't exist. The programmer should use `oldestAge` which they created in line 11



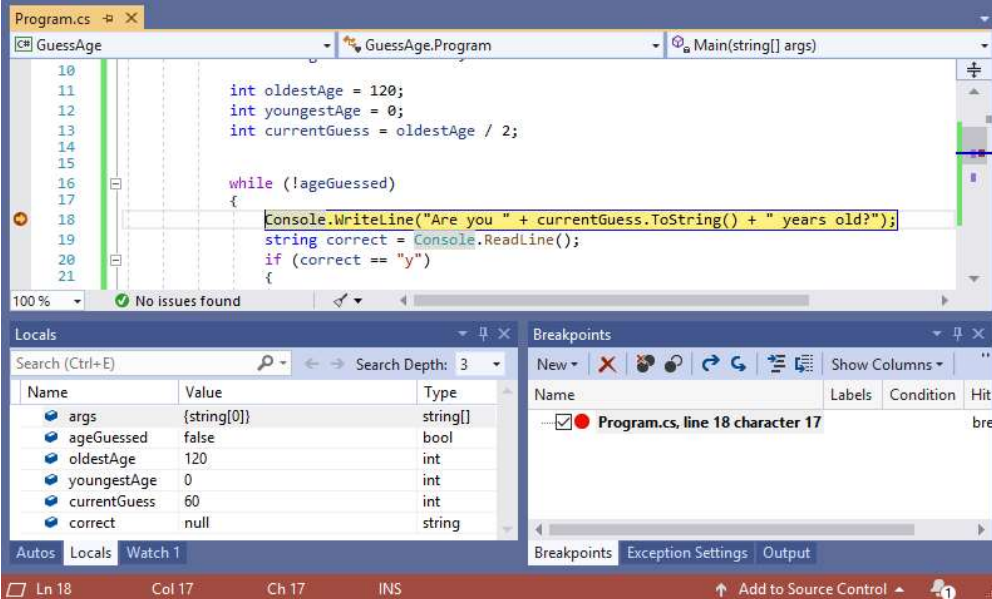
```
7  
8  
9  
10  
11 int oldestAge = 120;  
12 int youngestAge = 0;  
13 int currentGuess = oldstAge / 2;  
14  
15  
16 while (!ageGessed  
17 {  
18     Console.WriteLine("Are you " + currentGuess.ToString() + "  
19     string correct = Console.ReadLine());  
20     if (correct == "y")  
21     {
```

Error List

Code	Description	Project	File	Line
CS1026	) expected	GuessAge	Program.cs	16
CS1013	The name 'oldstAge' does not exist in the current context	GuessAge	Program.cs	13

# Debugging code

- Breakpoints are set by the programmer so that the IDE stops the program mid-way through running
  - The programmer can then step through code line by line
  - They can watch variables as they change
- The breakpoint was set to line 18
  - What do the variables contain at this point?



```
Program.cs [x]
[?] GuessAge
[?] GuessAge.Program
[?] Main(string[] args)
10
11
12
13
14
15
16
17
18
19
20
21
int oldestAge = 120;
int youngestAge = 0;
int currentGuess = oldestAge / 2;

while (!ageGuessed)
{
    Console.WriteLine("Are you " + currentGuess.ToString() + " years old?");
    string correct = Console.ReadLine();
    if (correct == "y")
    {

```

Name	Value	Type
args	{string[0]}	string[]
ageGuessed	false	bool
oldestAge	120	int
youngestAge	0	int
currentGuess	60	int
correct	null	string

Name	Labels	Condition	Hit
Program.cs, line 18 character 17			bre

# Variables

Answers

- `ageGuessed = false`
- `oldestAge = 120`
- `youngestAge = 0`
- `currentGuess = 60`
- `correct = null` (it hasn't yet been assigned a value)
- `args = {string[0]}`  
These are arguments sent to the program when it is called



The screenshot shows a 'Locals' window with a search bar and a table of variables. The table has three columns: Name, Value, and Type. The variables listed are args, ageGuessed, oldestAge, youngestAge, currentGuess, and correct, with their respective values and types.

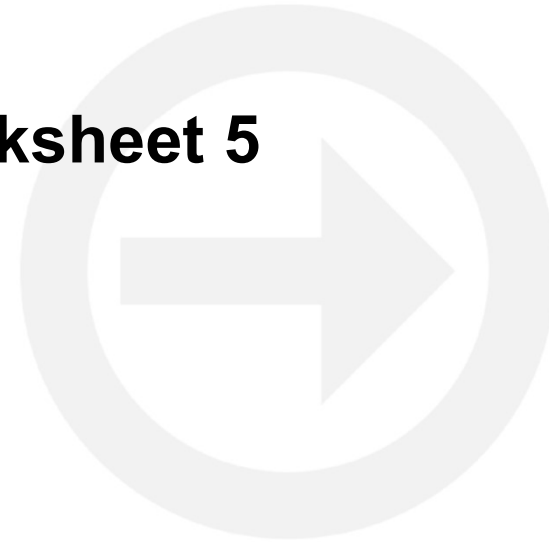
Name	Value	Type
args	{string[0]}	string[]
ageGuessed	false	bool
oldestAge	120	int
youngestAge	0	int
currentGuess	60	int
correct	null	string

# Run-time environment

- The run-time environment allows a programmer to test their program while it is running
  - If the program crashes the run-time environment can see what happened and give useful information to the programmer
- The run-time environment also means the libraries that come with the programming language and will be available to the programmer to use

# Worksheet 5

- Now complete **Task 2** on **Worksheet 5**





# Plenary

- With a partner, take it in turns to explain each of the following terms related to IDEs:
  - Editors
  - Line numbers
  - Syntax highlighting
  - Breakpoints, stepping and watching variables
  - Error diagnostics
  - Run-time environment

# Plenary

## Answers

- **Editors** – Use to write and edit programming code
- **Line numbers** – Give a number to each line of code, this is useful when the IDE refers to lines of code
- **Syntax highlighting** – Different colours for code that has different meanings, such as keywords, strings or variables
- **Breakpoints, stepping and watching variables** – The ability to pause code when running and then run each line of code separately whilst watching the values stored in variables as they change
- **Error diagnostics** – Comments from the IDE that help in finding the cause of syntax errors
- **Run-time environment** – The facility of the IDE that allows programs to be run



## Copyright

© 2020 PG Online Limited

The contents of this unit are protected by copyright.

This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it are supplied to you by PG Online Limited under licence and may be used and copied by you only in accordance with the terms of the licence. Except as expressly permitted by the licence, no part of the materials distributed with this unit may be used, reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic or otherwise, without the prior written permission of PG Online Limited.

## Licence agreement

This is a legal agreement between you, the end user, and PG Online Limited. This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it is licensed, not sold, to you by PG Online Limited for use under the terms of the licence.

The materials distributed with this unit may be freely copied and used by members of a single institution on a single site only. You are not permitted to share in any way any of the materials or part of the materials with any third party, including users on another site or individuals who are members of a separate institution. You acknowledge that the materials must remain with you, the licencing institution, and no part of the materials may be transferred to another institution. You also agree not to procure, authorise, encourage, facilitate or enable any third party to reproduce these materials in whole or in part without the prior permission of PG Online Limited.