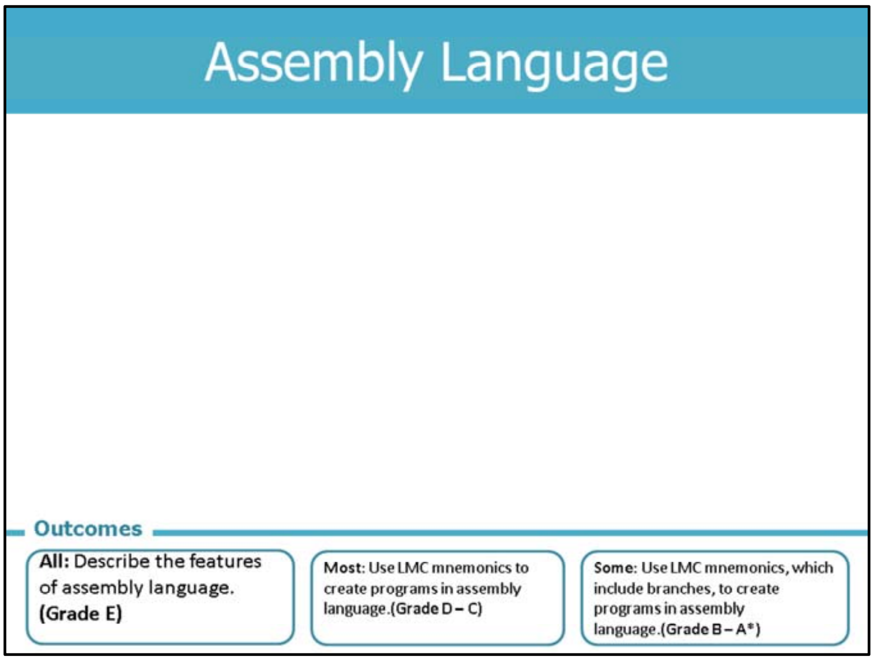


Assembly Language

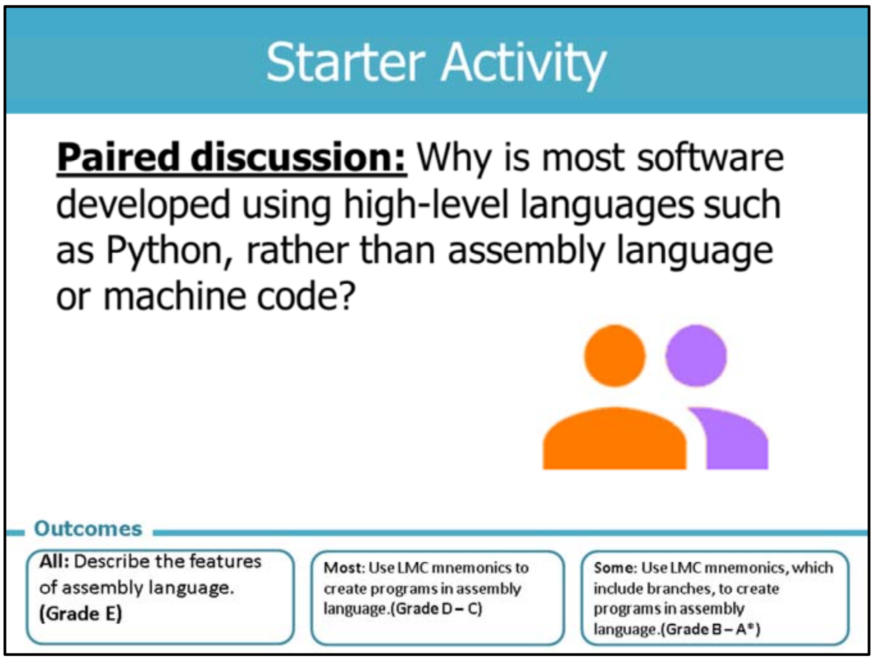
Outcomes

All: Describe the features of assembly language. **(Grade E)**

Most: Use LMC mnemonics to create programs in assembly language.**(Grade D – C)**

Some: Use LMC mnemonics, which include branches, to create programs in assembly language.**(Grade B – A*)**

Assembly language is a low-level programming language which uses mnemonics for the opcode and a symbolic name for the operand e.g. LDA firstOne



Starter Activity

Paired discussion: Why is most software developed using high-level languages such as Python, rather than assembly language or machine code?

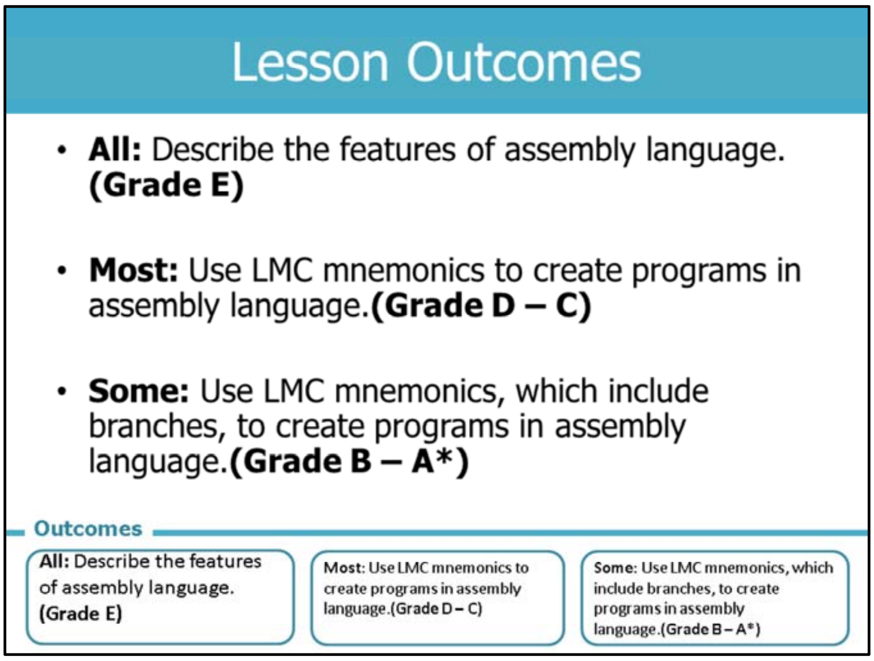
Outcomes

All: Describe the features of assembly language. (Grade E)

Most: Use LMC mnemonics to create programs in assembly language.(Grade D – C)

Some: Use LMC mnemonics, which include branches, to create programs in assembly language.(Grade B – A*)

…Python uses natural language so it's easier/more intuitive/focuses on problem solving rather than syntax
…it is portable so it can be compiled for many different CPU instruction sets



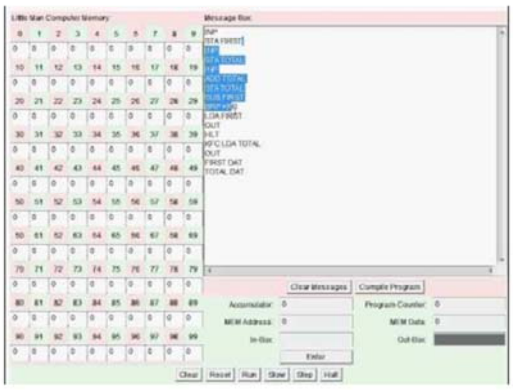
Lesson Outcomes

- **All:** Describe the features of assembly language. **(Grade E)**
- **Most:** Use LMC mnemonics to create programs in assembly language.**(Grade D – C)**
- **Some:** Use LMC mnemonics, which include branches, to create programs in assembly language.**(Grade B – A*)**

Outcomes

All: Describe the features of assembly language. **(Grade E)**	**Most:** Use LMC mnemonics to create programs in assembly language.**(Grade D – C)**	**Some:** Use LMC mnemonics, which include branches, to create programs in assembly language.**(Grade B – A*)**

Assembly Language



```
INP
STA FIRST
INP
STA TOTAL
INP
ADD TOTAL
STA TOTAL
SUB FIRST
BRP KFC
LDA FIRST
OUT
HLT
KFC LDA TOTAL
OUT
FIRST DAT
TOTAL DAT
```

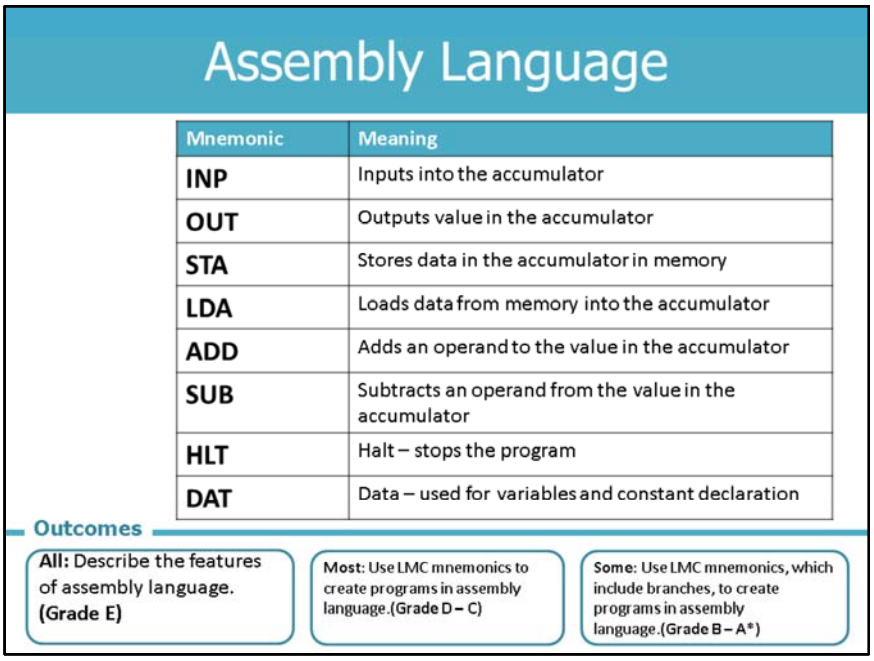
Outcomes

All: Describe the features of assembly language. **(Grade E)**

Most: Use LMC mnemonics to create programs in assembly language.(Grade D – C)

Some: Use LMC mnemonics, which include branches, to create programs in assembly language.(Grade B – A*)

Use the materials on the following site, including the Applet which you will need to launch:
http://www.yorku.ca/sychen/research/LMC/



Assembly Language

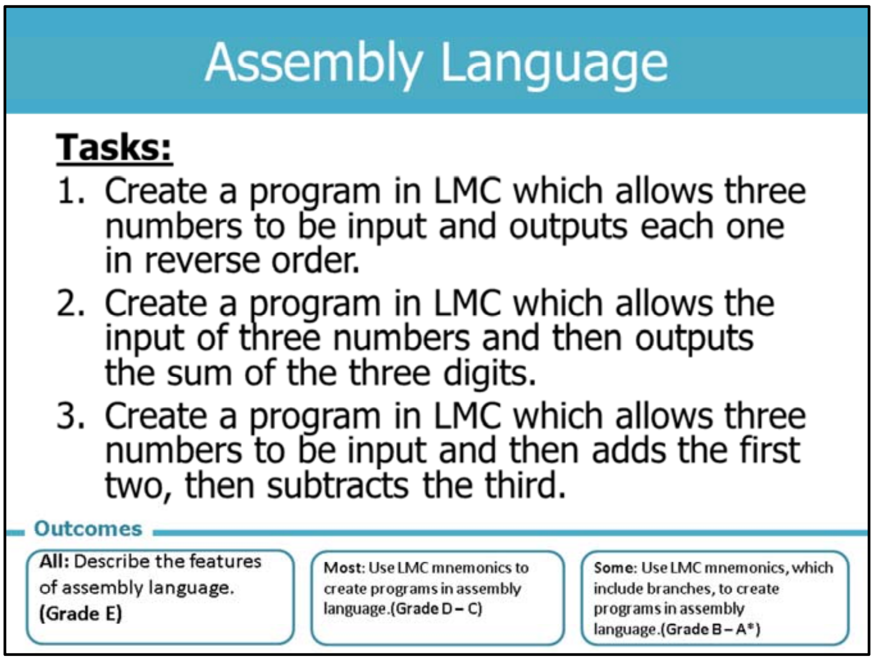
Mnemonic	Meaning
INP	Inputs into the accumulator
OUT	Outputs value in the accumulator
STA	Stores data in the accumulator in memory
LDA	Loads data from memory into the accumulator
ADD	Adds an operand to the value in the accumulator
SUB	Subtracts an operand from the value in the accumulator
HLT	Halt – stops the program
DAT	Data – used for variables and constant declaration

Outcomes

All: Describe the features of assembly language. **(Grade E)**

Most: Use LMC mnemonics to create programs in assembly language.**(Grade D – C)**

Some: Use LMC mnemonics, which include branches, to create programs in assembly language.**(Grade B – A*)**



Assembly Language

Tasks:

1. Create a program in LMC which allows three numbers to be input and outputs each one in reverse order.
2. Create a program in LMC which allows the input of three numbers and then outputs the sum of the three digits.
3. Create a program in LMC which allows three numbers to be input and then adds the first two, then subtracts the third.

Outcomes

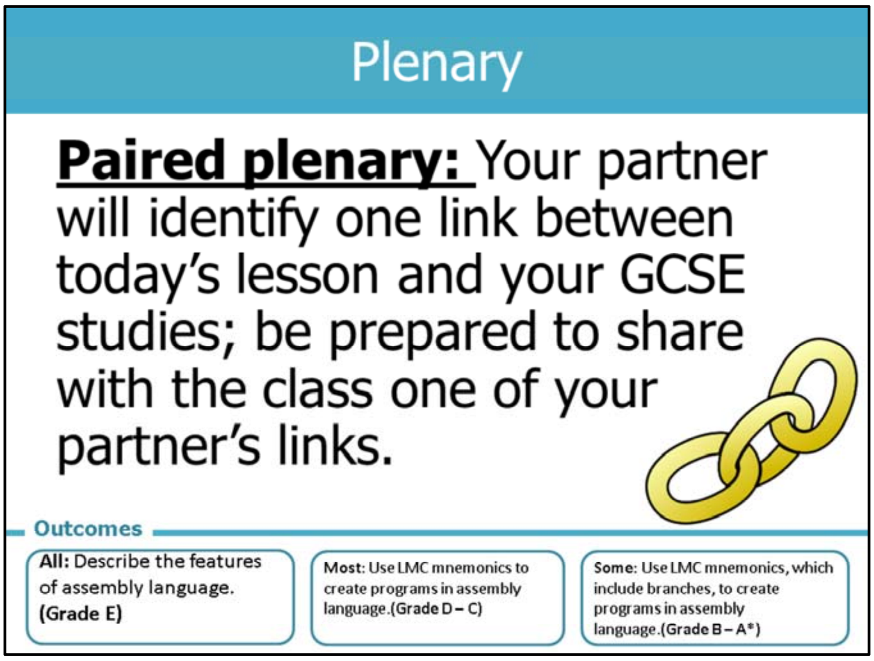
All: Describe the features of assembly language. (Grade E)	**Most:** Use LMC mnemonics to create programs in assembly language.(Grade D – C)	**Some:** Use LMC mnemonics, which include branches, to create programs in assembly language.(Grade B – A*)

An illustration, to get you started (the # is a comment and does not appear in the code)

1. EXEMPLAR SOLUTION FOR TASK 1

INP #input – this input will be sotred in the register, ACC. Notice how this doesn't have an operand)
STA num1 #STA stores value in the accumulator in memory as the next input will overwrite the first number entered into the ACC
INP #input – again this puts the value in the ACC overwriting the last input, which is now stored in memory
STA num2 #STA – this needs to be stored too as there is one more input to collect, we use a different identifier so it doesn't overwrite num1 in memory
INP
OUT #Output the value in the accumulator – i.e. the last input will be in there – it doesn't need to be stored, but could if you wanted
LDA num2 #LDA – load, operand num2 and put it in the ACC
OUT #output contents of the ACC

```
LDA num1                    #load the first number into the ACC
OUT                         #output it
HLT                         #halt the program
num1 DAT                    #declare variable/data
num2 DAT                    #declare variable/data
```



Plenary

<u>Paired plenary:</u> Your partner will identify one link between today's lesson and your GCSE studies; be prepared to share with the class one of your partner's links.

Outcomes

All: Describe the features of assembly language. **(Grade E)**

Most: Use LMC mnemonics to create programs in assembly language.**(Grade D – C)**

Some: Use LMC mnemonics, which include branches, to create programs in assembly language.**(Grade B – A*)**