# Topic Exploration Pack

# Types of Programming Language

# Activity 1

This website (<http://langpop.com/>) contains the ‘hit parade’ of programming languages. Copy out the
top 20 into the worksheet rows below and in the second column, following some research, indicate if a language is P (procedural), F (functional), O (object-oriented), L (low-level), M (multi-paradigm) or
A (all others) with a little extract of code that proves your verdict.

|  |  |
| --- | --- |
| **Language** | **Paradigm** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Then tally up the results:

|  |  |
| --- | --- |
| **Type** | **Number of languages in the table** |
| P (procedural) |  |
| F (functional) |  |
| O (object-oriented) |  |
| L (low level) |  |
| M (multi-paradigm) |  |
| A (all others) |  |

Which paradigm is the most popular then?

# Activity 2

The problem is given in structured English as follows:

‘Input two numbers x and y and check if x is exactly 1 larger than y; if it is then output 11 if yes, or 22 if false.’

Question 1: Create pseudocode for this program.

[Solution]

BEGIN numDiffOne

 INPUT x

 INPUT y

 IF x-y=1 THEN

 OUTPUT "11"

 ELSE

 OUTPUT "22"

 END IF

HALT

Question 2: Create a procedural implementation of this program.

[Solution]

Python:

def get\_data():

 x=input('? ')

 y=input('? ')

 return (int(x)-int(y))

def compare\_nums(result):

 if result==1:

 print("11")

 else:

 print("22")

def main():

 a=get\_data()

 compare\_nums(a)

main()

The above program produces this result:

>>> ================================ RESTART

>>>

? 7

? 5

22

>>> ================================ RESTART

>>>

? 7

? 6

11

>>>

Question 3: Create an LMC implementation of this program.

[Solution]

INP

STA x

INP

STA y

LDA x

SUB y

SUB one

BRZ tr

BRA fls

tr LDA true

BRA endif

fls LDA false

endif OUT

HLT

x DAT

y DAT

one DAT 1

true DAT 11

false DAT 22



# Activity 3

Describe how you would implement an object-oriented game of Twenty one (Blackjack). The program will generate up to four cards for the user drawn out from a deck, giving the user the option to stop when they think they got as close to 21 as possible. Then the program will generate the cards for itself, trying to exceed the user’s score but without going over 21. Picture cards are worth 10 and Ace can be worth either 1 or 11.

List the classes you will use in this game and provide a class diagram:

Describe in words any instances of inheritance and polymorphism in your program and justify the reason for usage.