```python
"""
CP1404/CP5632 Practical - Suggested Solution
Programming Language class with tests.
"""


class ProgrammingLanguage:
    """Represent information about a programming language."""

    def __init__(self, name, typing, reflection, pointer_arithmetic, year):
        """Construct a ProgrammingLanguage from the given values."""
        self.name = name
        self.typing = typing
        self.reflection = reflection
        self.pointer_arithmetic = pointer_arithmetic
        self.year = year

    def __str__(self):
        """Return string representation of a ProgrammingLanguage."""
        return f"{self.name}, {self.typing} Typing, Reflection={self.reflection}, " \

               f"Pointer Arithmetic={self.pointer_arithmetic}, First appeared in {self.year}"

    def is_dynamic(self):
        """Determine if language is dynamically typed."""
        return self.typing == "Dynamic"


def run_tests():
    """Run simple tests/demos on ProgrammingLanguage class."""
    ruby = ProgrammingLanguage("Ruby", "Dynamic", True, False, 1995)
    python = ProgrammingLanguage("Python", "Dynamic", True, False, 1991)
    visual_basic = ProgrammingLanguage("Visual Basic", "Static", False, False, 1991)

    languages = [ruby, python, visual_basic]
    print(python)

    print("The dynamically typed languages are:")
    for language in languages:
        if language.is_dynamic():
            print(language.name)


if __name__ == "__main__":
    run_tests()
```