|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10차시 | 1 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제1) 다음 오류 중 코드 **실행**에 영향을 미치지 않는 오류를 고르시오.  (1) 문법 오류  **(2) 논리 오류**  (3) 컴파일 오류  (4) 실행 오류  문제2) 리스트를 [1, 2)이라고 코딩하면, 어떤 오류가 발생하는가?  **정답2) 문법 오류 (Syntax Error)**  문제3) 다음 코드는 **1부터 10까지** 한 줄에 정수 한 개씩 출력하는 코드이다. 코드를 실행시켜보고, 오류가 있다면 그 종류를 설명하라. 또한, 오류가 발생했음에도 프로그램이 중간에 멈추지 않는 이유를 설명하고, 오류를 해결한 코드를 작성하라.   |  | | --- | | *Python* | | for i in range(1,10):      print(i) |   **정답3) 1부터 10까지 출력해야 하는데 1~9까지만 출력된다. 이는 논리 오류에 해당된다. 문법 또는 실행 오류가 아니므로 프로그램이 중간에 멈추지 않는다. range() 객체가 마지막 숫자를 포함시키지 않기 때문에 10까지 출력하고 싶다면 11까지 써야 한다.**  **for i in range(1,11):**  **print(i)**  문제4) 다음 코드에 문자열을 입력하면 발생하는 오류의 종류가 무엇인지와 해당 오류를 어떻게 해결해야 하는지를 서술하라.   |  | | --- | | *Python* | | n = int(input("정수를 입력하세요: ")) |   **정답4) 정수가 아닌 문자가 입력되면 실행 오류가 발생한다. 이런 오류가 발생할 때 해결하는 방법으로 두 가지 정도를 생각할 수 있다.**  **첫 번째는 input() 함수가 반환하는 값이 숫자로만 구성되었는지 미리 확인하는 것 이다. 문자열의 isdigit() 함수 같은 것들이 유용하게 사용될 수 있다. 미리 확인하고 오류가 발생할 것 같으면 int() 함수를 호출하지 않고, 사용자에게 다시 입력하라고 할 수 있다.**  **두 번째는 input() 함수를 호출해서 입력 받은 내용을 int() 함수에 전달한 뒤 예외 처리를 하는 것이다.ValueError 등이 발생하면 이를 처리하는 코드에서 다시 입력하도록 할 수 있다.** | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 문제5) 다음 오류의 종류를 구분해본다.  (1) 원 둘레를 계산하는 공식의 오류  (2) 삭제된 파일을 읽는 오류  (3) 0으로 나누는 오류  (4) 곱하기 부호를 'X'로 쓴 오류  (5) 리스트의 요소 개수보다 큰 범위를 접근하려는 오류  (6) 파일을 쓰다가 디스크 공간이 없는 오류  **정답5)**  **(1) 논리 오류**  **(2) 실행 오류**  **(3) 실행 오류**  **(4) 문법 오류**  **(5) 실행 오류**  **(6) 실행 오류**  문제6) 사용자로부터 입력을 받는 경우에는 오류가 발생하기 쉽다. 아래 제공된 코드에서 이를 해결하기 위해 try / except 구문으로 예외 처리 코드를 작성하라. 단 예외가 발생할 경우에는 “숫자를 입력하지 않았어요” 라고 출력하게 만들어라.   |  | | --- | | *Python* | | s = input("정수를 입력하세요: ")  n = int(s) |   **정답6)**  **try:**  **s = input("정수를 입력하세요: ")**  **n = int(s)**  **except:**  **print("숫자를 입력하지 않았어요")**    문제7) try / except 구문을 이용해 오류를 방지하는 것과, 아래 코드와 같이 if문을 이용해 오류를 방지하는 방법의 차이를 간단하게 서술하시오.   |  | | --- | | *Python* | | if not s.isdigit():      print("숫자를 입력하지 않았어요") |   **정답7)**  **결론적으로, try / except 구문은 예외 처리를 통해 오류에 대응하는 반면, if문은 조건 로직을 통해 오류 발생을 사전에 방지하는 방식을 취합니다.**  **if문은 주로 예측 가능한 조건에 대해 미리 검사를 하여 오류가 발생하지 않도록 방지하는 데 사용되며, try/except 구문은 예측 불가능한 상황에서 발생할 수 있는 예외를 처리하고, 프로그램의 안정성을 유지하기 위해 사용됩니다. 따라서, 각 상황에 맞게 적절히 선택하여 사용하는 것이 중요합니다.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10차시 | 8 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제8) 현재 실행되고 있는 프로그램을 강제로 종료하는 것은 좋지 못하다. 프로그램을 죵료하는 코드를 작성하라.   |  | | --- | | *Python* | | import sys  # write your code here |   **정답8) 코드가 실행하는 중에 프로그램을 종료하는 방법은 sys모듈의 exit()함수를 사용 하는 것이다.**    **import sys**  **# 작업을 함**  **# 종료하고 싶을 때 sys.exit() 함수 호출**  **sys.exit()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10차시 | 9 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제9) 사용자가 입력한 텍스트 파일을 읽고 화면에 출력하는 프로그램을 작성하되, 파일이 없는 경우에 "파일이 존재하지 않습니다"를 출력하고 종료한다.  **정답9)**    **import sys**  **fileName = input("파일 이름을 입력하세요: ")**  **try:**  **f = open(fileName)**  **except FileNotFoundError:**  **print(f"{fileName}은 없는 파일입니다")**  **fileName = input("파일 이름을 다시 입력하세요")**  **try:**  **f = open(fileName)**  **for line in f:**  **print(line)**  **f.close()**  **except FileNotFoundError:**  **print(f"{fileName}이 없습니다. 프로그램을 종료합니다.")**  **sys.exit()** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10차시 | 10 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제10) 다음 두 print()문에 대한 예외 처리를 구현해 다음과 같은 결과가 나오도록 아래 코드를 완성하시오..  (1) print(int(“abc”))에 대한 결과  예외 발생 이름: <class 'ValueError'>  예외 발생 이유: invalid litera; for int( with base 10: ‘abc’)  예외 처리가 잘되는군요!   |  | | --- | | *Python* | | try:      print(int("abc"))  except Exception as e:      # write your code here  else:      # write your code here  finally:      # write your code here |   **정답10)(1)**  **try: # (1)**  **print(int(oct(10)))**  **except Exception as e:**  **print(f'예외 발생 이름: {type(e)}')**  **print(f'예외 발생 이유: {e}')**  **else:**  **print('잘 실행됬습니다.')**  **finally:**  **print('예외 처리가 잘되는군요!')**    (2) print(“10”)에 대한 결과  12  잘 실행됐습니다.  예외 처리가 잘되는군요!   |  | | --- | | *Python* | | try:      print(int("10"))  except Exception as e:      # write your code here  else:      # write your code here  finally:      # write your code here |   **정답10)(2)**  **try: # (2)**  **print(int(oct(10)[:2]))**  **except Exception as e:**  **print(f'예외 발생 이름: {type(e)}')**  **print(f'예외 발생 이유: {e}')**  **else:**  **print('잘 실행됬습니다.')**  **finally:**  **print('예외 처리가 잘되는군요!')**    문제11) 함수 divide(x, y) 함수를 구현 한 후 연산 0으로 나누는 것에 대한 예외 처리를 수행하도록 한다.  - 구문은 try, except, else 블록 구현  - 다음 두 호출에 대한 결과  결과: 1.6  0으로는 나눌 수 없습니다.   |  | | --- | | *Python* | | def divicde(x, y):      answer = x / y        return answer |   **정답11)**  **def divide(x, y):**  **try:**  **answer = x / y**  **except ZeroDivisionError:**  **print('0으로는 나눌 수 없습니다.')**  **else:**  **print(f'결과: {answer}')**    **divide(3.2, 2)**  **divide(5.4, 0)**  문제12) 존재하지 않는 파일을 읽으려 할 때 발생하는 예외를 처리하고, 파일이 성공적 으로 읽혔는지 여부에 따라 메시지를 출력하세요.  - try, except(file not found) 블록을 사용할 것  **정답12)**  **def read\_file(filename):**  **try:**  **file = open(filename, 'r', encoding="utf-8")**  **content = file.read()**  **except FileNotFoundError:**  **print("파일을 찾을 수 없습니다.")**  문제13) 사용자로부터 입력 받은 문자열을 정수로 변환할 때 발생할 수 있는 예외를 처리하세요.  - try, except(value error) 블록을 사용할 것  **정답13)**  **def convert\_to\_int():**  **try:**  **user\_input = input("정수로 변환할 문자열을 입력하세요: ")**  **result = int(user\_input)**  **except ValueError:**  **print("입력한 문자열을 정수로 변환할 수 없습니다.")** | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10차시 | 14 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |
| 문제14) 사용자로부터 두 개의 숫자와 연산자를 입력받아 사칙연산(+, -, \*, /)을 수행 하는 프로그램을 작성하세요. 잘못된 입력이나 연산에서 발생할 수 있는 예외를 처리하고, 계산이 성공했는지 여부에 따라 메시지를 출력하세요.     |  | | --- | | *Python* | | def calculator():      try:          num1 = float(input("첫 번째 숫자를 입력하세요: "))          num2 = float(input("두 번째 숫자를 입력하세요: "))          operator = input("연산자를 입력하세요 (+, -, \*, /): ")          # write your code here | |  |   **정답14)**  **def calculator():**  **try:**  **num1 = float(input("첫 번째 숫자를 입력하세요: "))**  **num2 = float(input("두 번째 숫자를 입력하세요: "))**  **operator = input("연산자를 입력하세요 (+, -, \*, /): ")**  **# write your code here**  **if operator == '+':**  **result = num1 + num2**  **elif operator == '-':**  **result = num1 - num2**  **elif operator == '\*':**  **result = num1 \* num2**  **elif operator == '/':**  **result = num1 / num2**  **else:**  **raise ValueError("유효하지 않은 연산자입니다.")**    **except ValueError as e:**  **print(f"입력 오류: {e}")**  **except ZeroDivisionError:**  **print("0으로 나눌 수 없습니다.")**  **else:**  **print(f"결과: {result}")**  **finally:**  **print("계산이 완료되었습니다.")** | | | | | | |