



4.1.1 Exception Handling

1) Explain what exception handling is used for.

[2 marks]

- To stop a program from crashing;
 - To deal with (an anticipated) run-time error; (Accept a clear description of an example of a run-time error)
 - To avoid the structure of code becoming too complex as a result of trying to check for potential errors before they might occur;
-

2) State the purpose of each of the following Python keywords:

a) try

[1 mark]

- Contains code that might cause an exception;

b) except

[1 mark]

- Contains code that runs when a specified exception occurs;

c) finally

[1 mark]

- Contains code that runs whether or not an exception occurs.;
-





3) Consider the program shown in Figure 1 that takes an input from the user.

Figure 1

```
number = int(input("Enter a number: "))
result = 100 / number
```

a) State the name of two errors that could occur during the execution of this program and explain how each could occur.

[4 marks]

- **ValueError;**
- **if the user enters something that cannot be converted to an integer, such as "hello".**
- **ZeroDivisionError;**
- **if the user enters 0, because the program attempts to divide 100 by zero;**

b) Modify the program to use exception handling to handle these errors.

[2 marks]

- **try:**
 number = int(input("Enter a number: "))
 result = 100 / number
except ValueError:
 print("Please enter a valid number.")
except ZeroDivisionError:
 print("You cannot divide by zero.")

Marking points:

Uses try to contain the code that could generate an exception. [1]

Correctly handles the possible exceptions using appropriate except blocks. [1]

