****

1.12: Worksheet 1

For each of the questions which require you to write code, you should add your answers to the Python file you started within the practical session. It will be a good idea if you were to put a comment above stating which question you are answering. Comments are started with a hashtag (#):



Work your way through these questions at your own pace. If you need some help after trying to find the answer for yourself, please ask your module tutor.

1. Complete the following table relating to programming paradigms:

| **Paradigm** | **Description** | **Positives** | **Negatives** | **Examples** |
| --- | --- | --- | --- | --- |
| Imperative | It performs step by step task by changing state. Its focus is on how to achieve the goal, the paradigm consists of several statements and after execution of all, the results is stored. | 1. Very simple to implement 2. It contains loops, variables, etc. | 1. Complex problems cannot be solved. 2. Less efficient and less productive. | 1. C; by Dennis Ritchie and Ken Thompson 2. Fortran; by John Backus 3. Basic; by G J Kemeny and T E Kurts |
| Procedural | It emphasises on procedure, in terms of underlying machine model.  This involves writing down a list of instructions to tell the computer what to do in a step by step to finish the task at hand. Predefines functions, local variables, modularity, parameter passing. | 1. The program flow can be tracked easily. 2. Excellent for general purpose programming. 3. Has the ability to reuse the code. | 1. Program code is harder to write. 2. Difficult to relate to real world objects. 3. Data is exposed to the whole program, making it not so much security friendly. | 1. Fortran. 2. COBOL 3. PASCAL 4. ALGOL |
| Object-oriented | OOP can encapsulate, each object in the program is self-sustainable, meaning all the components that make up the object are within the object itself.  Objects can be taken from one program and used to resolve another problem at hand with little or no alterations. | 1. Due to modularity and encapsulation, OOP offers ease of management. 2. OOP mimics the real world making it easier to understand. 3. Since objects are whole within themselves, they are reusable in other programs. | 1. Programs being built with OOP may take longer to be created 2. Over generalisation 3. It tends to be slower and uses high amount of memory | 1. Java 2. C++ 3. C# 4. Python 5. R 6. JavaScript 7. Ruby |
| Declarative | It is a style that expresses logic of computation without its control flow. The focus is about what needs to be done rather than how it should be done.  It basically emphasises on what the code is doing. | 1. Short efficient code. 2. Could be implemented using methods not known yet at the time of programming. | 1. Sometimes hard to understand for external people. 2. For others, it is based on an unfamiliar conceptual model (Solution State) | 1. HTML 2. XML 3. CSS 4. SQL 5. Prolog 6. F# 7. Lisp |
| Logic | Could be termed as abstract model of computation.  Main emphasis is on knowledge base and the problem.  It solves logical problems like puzzles, series, etc. | 1. Classified as high-level language as it implements logic computations rather than mechanics. 2. Allows data to be represented both extensionally and intentionally. | 1. The programs execution can be slow. 2. True/False statements cannot solve most problems at all. 3. Limited to which types of problem it can efficiently solve. | 1. Prolog |
| Functional | It is the oldest programming paradigm. Functional programming also known as applicative programming and value-oriented programming*.* Basically, functional programming is a declarative type of programming. In functional programming, everything is a function returning values instead of modifying data. | 1. Mutability makes the function code free of side effect. 2. Variables can be replaced by their values since the evacuation of expression can be done anytime. | 1. Since there is no state and no update of variables allowed, loss of performance will take place. | 1. JavaScript 2. Haskwell 3. Erlang 4. Lisp 5. Clojure |
| Database Processing | This is based on data and its movement. Program statement are defined by data rather than hard coding a series of steps.  It is the heart of a business information system and provides file creation, data entry, update, query, and reporting functions. | 1. Reduces data redundancy. 2. Improved data security 3. Reduces updating error and increases consistency. | 1. Damaged to database will affect virtually all applications programs. 2. Substantial hardware and software start-up costs. | 1. Microsoft SQL 2. MySQL 3. Amazon Relational Database Service (RDS) 4. ORACLE RDBMS 5. Knack |

1. As you have seen both in the previous question and in the class, programming paradigms each have their own uses. Investigate which paradigms are used most within industry and how they are utilized.

Answer: The most popular programming paradigms in the world of software development are, Object Oriented Programming, Functional Programming, Imperative Programming and Declarative Programming.

But the most used amongst them is the Object-Oriented Programming paradigm as its unique advantage like the modularity of code and ability to directly associate real world business problems in terms of code. Objects are instances of classes; it has attributes and methods.

Though other programming paradigms exist, the pure implementation of any one paradigm in a programming language is very difficult to find. Most languages are multiparadigmatic in nature.

1. Another type of Software Development Tool are frameworks and libraries.
   1. In terms of software development, what is a framework?

Answer: A framework inverts the control of the program. It tells the developer what they need. It is an abstraction in which software providing generic functionality can be selectively change by additional user written code, thus providing application-specific software.

* 1. In terms of software development, what is a library?

Answer: Unlike framework, the programmer calls the library when and where they need it.

Library is a collection of non-volatile resources used by computer programs, often for software development. These may include configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values, or type specifications.

* 1. How are frameworks and libraries used within software development?

Answer: In software development, the programmer calls a set of functions from the library code and passes the characters whose position they need to find as a parameter in the function call. Instead the programmer uses the framework’s code to call his code.

* 1. What are the differences between frameworks and libraries?

Answer: The key difference between a library and a framework is "Inversion of Control". When you call a method from a library, you are in control. But with a framework, the control is inverted: the framework calls you. ... The classes and methods normally define specific operations in a domain specific area. (programcreek.com, 2011)

* 1. What are the benefits of frameworks and libraries?

Answer: Frameworks and libraries help you build projects quicker by letting you write code in a language that adds superpowers to the way you build projects. Now there are libraries that extend the capabilities of existing languages, like “Lodash”, which helps you manage objects and arrays more easily.

Frameworks help you build projects quicker by giving you a structure that makes development more efficient.

* 1. What are the drawbacks of frameworks and libraries?

Answer: The framework’s core behavior cannot be modified, indicating that when you utilize a framework, you are required to respect its limitations and work the way it is required.

Since the framework is readily available to everyone, it can be studied to know how things work and to discover flaws that can be utilized against you. You learn the framework and not the language.

You should know exactly what the library function does, some of them will alter the values of the input variables, better known as side effects. It may not do exactly what you want it to do.

* 1. On balance, do you think the benefits outweigh the drawbacks?

Answer: yes, the benefits outweighs the drawbacks, it makes the work easy and time efficient.

* 1. Name at least three frameworks for Python and discuss their uses

Answer: (a). CherryPy: Allows building web applications in as much the same way one would build any other object-oriented program.

(b). Flask: Highly suitable for developers that want to make a standalone app.

(c). Django: Popular python framework, it follows the model-template-views architectural pattern.

* 1. Name at least three libraries available in Python and discuss their uses.

Answer: (a). Pillow: python image library

(b). Matplotlib: Uses python script to write two-dimensional graphs and plots.

(c). NumPy: It adds to python a large collection of high-level mathematical functions.

1. Enter the following code into your Python file, then run it:

print("123.456789 just using 0 is: {0}".format(123.456789))

print("123.456789 just using 0:0.2f is: {0:0.2f}".format(123.456789))

* 1. What do you think the {0:0.2f} part does?

Answer:The number of spaces after the “colon”

* 1. Change this to {0:10.2f}, what is happening now?

Answer: The number of spaces after the “colon” increased.

* 1. If you were required to print a number to 3 decimal places, how would you do this using the format command?

Answer:

print("190.7639475 is = {0:0.3f}".format(190.7639475))