Assignment Module 2 – Kotlin Programming Basics

1. Explain the different data types available in Kotlin.

Available Data Types in Kotlin :

- Char :
  - It used to store single Character.
  - Size of char is 1 byte.

- Int :
  - It used to store whole numbers.
  - Size of int is 4 bytes.

- Float :
  - It used to store Decimal Point numbers.
  - Size of float is 4 bytes.

- Double :
  - It used to store long range numbers or decimal numbers.
  - Size of double is 8 bytes.

- String :
  - It is used to store group of character.
  - Size of string is depending on char datatype.
  - It occupies plus 1 size relate to char.

- Boolean :
  - It is used to store true or false.
  - Size of Boolean is 2 bytes.

- Array :
    - It is used to store more than one value of same data type.
    - Size of array is depending of stored elements.

- Except this data type, there is more data types in Kotlin…

2. How do val and var differ?

Difference between val and var :

- Val is used to store constant value.
- Var is used to store changeable value.

3. What is a lambda expression in Kotlin, and where can it be used?

- Lambda expression is also known as anonyms function.
- We can define that which kind of operation will be perform on that function.
- We can put the body of function in parameters.
- Arrow (->) operator is use to separate body and arguments of function.
- Arguments are defined before arrow (->) operator.
- Body is defined after arrow (->) operator.

4. Describe the principles of Object-Oriented Programming (OOP).

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction.

5. Explain the differences between abstract class and interface in Kotlin and provide examples of when to use them.

- Abstract class
    - For abstract class, it must have at least one pure virtual function.
    - No need to create an object of abstract class.
    - Derived class in which abstract class derived, object of that derived class will call member function and data member of abstract class.
    - Abstract class also can not make a derived class or sub class.
    - Abstract class can not initialize value, it is a blueprint of another class.
    - Example : If we want to define a function without declaration, it means if we no need to write a body part of particular function, only need to define that, we can create an abstract class with pure virtual function…

- Interface
  - Interface is similar as a class.
  - We can us it instead of class.
  - We can also define and declare variable and method within interface.
  - Interface cannot take a value of another class. It means it can not become sub class/child class/derived class.
  - We can inherit interface of another class.
  - More than one interface can inherit simultaneously.
  - **Example :** When we want to take more than one parent class for inherit. But no concept of inherit more than one class in Kotlin, so instead of these we can create more than one interface to inherit more than one parent class to inherit…