|  |
| --- |
| ## Questionnaire Week 1: |
|  | 1 What is the difference between var and let?  Var is a constant we can change the value if we need to change it during the process of code , for example like numeral values , otherwise let means constant immutable you cannot change the value like name or values than never change. |
|  | 2 What is an optional?  An optional is a value than can be nil (null) and is representing by ? , this values show us when your value can have the proper value exam (optional 233 ) or Nil (no value). |
|  | 3 What is optional chaining vs optional binding?  Optional binding is used to check whether an optional variable or constant has a non-nil value, and if so, assign that value to a temporary, non-optional variable.  Optional chaining allows us to call properties, methods, and subscripts on an optional that might be nil. If any of the chained values return nil, the return value will be nil. It does not store the value on the left into a variable. |
|  | 4 What are the different ways to unwrap an optional? How do they work? Are they safe?  We have to options to unwrap an optional we use one to force it if we know that value exit we use ! at the end to force it , and we use If let to asking for first is exist value print if no print Nil, example  Optional unwrap knowing as a bad practise and no safe because you are forcing to have a value but that value could change and convert to Nil during the project , so the way using ! at the end to confirm to the machine we have a value and is not a Nil so we force for sure to show the value , the optional binding is a good practice , if we want to use we need to add if let , for example  Var OptionalBinding : Int? //we create optional value  OptinalBinding = 12 // we know that optional has a value  If let Namevalue = OptionalBinding // we use If let and we give a name (any) and we print the value.  {print (Namevalue) } |
|  | 5 What is a closure?  Is a block of code we can call later, this can help us to have more clear code and easy to read. We define a closure like this.  Let MiNewClosure: (String, Int) -> (Void) { (here the syntaxis we want to create in this closure)  }  Is similar than a functions. |
|  | 6 What is the difference between a class and a struct?  Class are reference types, 2 references point to the same reference. That means if the value change in a value that value will change in both.  A=5->  B=5->  B=A  Both are part of the same reference and for example in B we change the value A will change too. B->8, this change A->8 too.   * In the class we need to create a init always. * Class use heap (allocation of memory slower) * You can modify values. * You can use Inheritance.   Struct are value types of instances keep unique copy of data.  In this case for example  A=5-> A=5  B=5-> now change the value  B=A  In this case the are individuals values if we change B this is the only value, we change but no the value in A  B=5 -> B=8  A=5 -> A=5   * In this case init is created by default. * The struct use stack (lifo) (allocation memory in CPU faster). * By default, all is immutable so if we want to made changes we need to use this word (mutating). * They don’t have inheritance. |
|  | 7 What is the syntax '??' do?  Is another way to handle optional values is to provide a default value using the ?? operator. If the optional value is missing, the default value is used instead.  Exam,  let name = “Bryan”  let surname = “Almeida”  print “My name and surname is : \(name ?? surname)” |
|  | 8 What is a tuple?  Is a list to ordinate elements we use it to call elements included in that list. Is a list to return values always we going to receive a value.For exam  Let MiNewTuple = (“Bryan”, 32 , “London”) // to call the tuple  We use the name . and we can call the data in the list we create by position on the list  Print MiNewTuple.0  Print MiNewTuple.1  Print MiNewTuple.2 |
|  | 9. What is Any vs AnyObject?  We use Any when we don’t know the value type so we use Any we can represent any instance of any type include function types and optional types.  We use Anyobject when we want to represent reference types basically. |
|  | 10. What is a protocol?  Is an interface that define methods, properties and other requirements that respond to a particular task. It can be adopted by a class, structure or enumeration with the aim of providing an implementation to these requirements. Any type that implements a certain protocol is said to implement (or conform to) that protocol. Key word protocol  Exam.  protocol MynewProtocol {  here we create the values (functions)  }  If we want to use it in our class, or struct or Enum,  Class miNewClass : MynewProtocol {  } |
|  | 11. What is Delegation  Is important concept this used to share or delegate the same actions between classes or structs this action works close in protocols that will be delegated in such a way that it is guaranteed that the type that implements this protocol (known as delegate) will provide the functionalities that have been delegated. |