Task 1

Imperative Programming as the name suggests is a type of programming paradigm that describes how the program executes. Developers are more concerned with how to get an answer step by step. It comprises the sequence of command imperatives. In this, the order of execution is very important and uses both mutable and immutable data. Fortran, Java, C, C++ programming languages are examples of imperative programming.

Declarative Programming as the name suggests is a type of programming paradigm that describes what programs to be executed. Developers are more concerned with the answer that is received. It declares what kind of results we want and leave programming language aside focusing on simply figuring out how to produce them. In simple words, it mainly focuses on end result. It expresses the logic of computation. Miranda, Erlang, Haskell, Prolog are a few popular examples of declarative programming.

Task 2

Null pointer exception

Primitives: variables that contain data. If you want to manipulate the data in a primitive variable you can manipulate that variable directly. By convention primitive types start with a lowercase letter. For example variables of type int or char are primitives

Task 3

من اللغات التى تعتبر غير حساسه هي

Haskell , prolog , Go

Task 4

Difference between Heap and Stack

**لكل برنامج في اللغات المذكورة آنفاً مكانين يتم فيها حفظ البيانات و التعليمات خلال فترة عمل البرنامج و تسمى هذه الأمكنة (Stack) و (Heap). عند بدأ أي برنامج (Process Instance) أو (Program Thread) سوف يتم إنشاء (Stack) خاص بها بينما (Heap) يعتبر مكان مشترك بين البرامج، مثلاً في لغة الجافا بيئة (JVM) تحتوى على (Heap) واحد فقط ويتم استخدامة من جميع برامج الجافا التي تعمل تحت هذه (JVM) بينما لكل برنامج (Stack) واحد بفتراض انه يستخدم (Single Thread).  
  
  
ببساطة يمكن القول أن (Stack) يقوم بتسجيل سير عمل البرنامج بحيث متى ما تم إستدعاء للدالة فإنها سوف توضع في (Stack) للتنفيذ. دعونا نعمل مقارنة بسيطه بين (Stack) و (Heap):  
  
  
يتضح لنا الآن قاعدة بسيطة:  
- جميع (Reference Types) مثل الكائنات يتم حفظها في (Heap)  
- أما بالنسبة للمتغيرات (Value Types) و المؤشرات (Pointers) فيتم حفظها في المكان التي عرفت فيه. بمعنى أنها أحيانا تكون في (Stack) و أحيانا تكون في (Heap))**