Exam 2 Study Guide:

Object Oriented Programming:

What are the differences between procedural and object oriented programming languages?

Define the following terms:
- Class
- Object
- Method
- Request
- Attribute
- Inheritance
- Interface
- User Interface
- Variable
- Program flow

Programming in Avenue

Document the following code. Create a header file listing all variables, and the type of object that each variable references. Then document each line of code.

```avenue
myView = av.FindDoc("View1")
myThemes = myView.GetThemes
loop = true
while (loop)
  firstTheme = MsgBox.ListAsString(myThemes, "Select one theme",""
  secondTheme = MsgBox.ListAsString(myThemes, "Select another theme",""
  firstFTab = firstTheme.GetFTab
  secondFTab = secondTheme.GetFTab
  firstThemeType = firstFTab.GetShapeClass.GetClassName
  secondThemeType = secondFTab.GetShapeClass.GetClassName
  if (firstThemeType = secondThemeType) then
    loop = false
  else
    if (MsgBox.YesNo("Your theme types are not the same.", "Try again?",true).Not
      exit
      end
    end
    newFileName = "c:\newfile.shp".AsFileName
    newFTab = FTab.MakeNew(newFileName, firstFTab.GetShapeClass)
    newTheme = FTheme.Make(newFTab)
    idFld = Field.Make("ID", #FIELD_DECIMAL, 8, 0)
    idFld.SetVisible( TRUE )
    newFTab.AddFields({idFld})
    myView.AddTheme(newTheme)
    newFTab.SetEditable(true)
    newShapeField = newFTab.FindField("Shape")
  endShapeField = firstFTab.FindField("Shape")
  for each rec in firstFTab
    newRec = newFTab.AddRecord
    oldShape = firstFTab.ReturnValue(firstShapeField, rec)
    newFTab.SetValue(newShapeField, newRec, oldShape)
    newFTab.SetValue(idFld, newRec, newRec)
  endShapeField = secondFTab.FindField("Shape")
  for each rec in secondFTab
    newRec = newFTab.AddRecord
    oldShape = secondFTab.ReturnValue(secondShapeField, rec)
    newFTab.SetValue(idFld, newRec, newRec)
  end
newFTab.SetEditable(false)
```
Object Models:

What are the different types of multiplicity relationships that you can have in an object model? Which of these are shown in the diagram below?

What is a generalization relationship? Are there any in the diagram below?

What is an aggregation relationship? Are there any in the diagram below?

There are two classes in the below diagram that could be abstract. Which are they?

Can you have a relationship that is both a generalization and a one-to-many relationship?

Can you have a generalization relationship that is also an aggregation relationship?

Can you have a relationship that is both aggregation and one-to-many?

In the box below, draw a diagram, similar to those I drew in class, showing the structure outlined in the object model diagram above.

Table