**ITEC 1505 Fundamentals of Programming – Fall 2023**

**Final – Due Monday, December 11, 2023.**

**Name:**

|  |  |
| --- | --- |
| **#** | **Chapter 9 – Python Exception Handling** |
| 1 | What is the difference between a compiler and an interpreter? *A compiler and an Interpreter are both high level master programs for verifying the syntax of a program file. The difference is that an Interpreter executes one statement of code at a time while a Compiler validates the calls to internal libraries and any API and runs the code all at once.* |
| 2 | What is a Python syntax error? Provide an example. *A Python Syntax Error occurs when an Interpreter encounters an invalid syntax in the python code. Examples of this would be a misspelled word or a missing closing bracket.* |
| 3 | What is the difference between a semantic error and a logical error? *A Semantic error occurs when there is a mistake in the code, such as misspelled words or variables. This will cause the Interpreter to flag an error. A logical error occurs when the code successfully runs, but there is a mistake in the code not caught by the Interpreter, such as incorrect math (ie. 4+6\*10=100).* |
| 4 | What’s the difference between file handling & exception handling? Provide example *A File handling error occurs when a file called for does not match the actual file name. Exception handling is a way to subvert a problem when the file tries to execute, and to keep the program from crashing.*  *An example would be to use the “try” function to open a file, and if that doesn’t work, you would have an “except” function to print that the file was not found.* |
| 5 | What is the purpose of the **try/except** commands? Provide an example. *The try/except commands are exception handling commands for file handling errors.*  *An example of this would be try: result = 4/0*  *except: ZeroDivisionError: print(“Cannot divide by 0)* |
| **#** | **Chapter 10 – Python Classes and Objects** |
| 6 | What is Object Oriented Programming? *OOP is a programing paradigm that uses the concepts of a class and related instances. The idea is to have a class as an overall idea and have varying objects (attributes) that reside within that class.* |
| 7 | Classes provide the means of bundling attributes and method functionality. True/False  *True* |
| 8 | What is a Class constructor? Provide an example. *Class Constructors are used for creating objects. As an example, the default class constructor is-*  *def\_\_init\_\_(self):* |
| 9 | What is the difference between a function and a method? *A function is a piece of code that can be called by name, and used throughout the program. A method is a piece of code that can be called by name, but is always associated with an object.* |
| 10 | Create a class Employee. Then, create an employee-object from it.  *class Employee(): #class*  *def\_\_init\_\_(self, name)*  *Emp1 = Employee (“Sam”) #employee-object* |
| **#** | **Chapter 11 – Python Class Inheritance** |
| 11 | Create a subclass Manager from a superclass Employee  *class Employee:*  *def \_\_init\_\_(self, first, last):*  *self.name=first*  *self.last=last*  *class Manager(Employee):*  *pass* |
| 12 | What is class inheritance? Provide an example. *Class Inheritance is the use of adding the attributes of a superclass to a subclass, as in the above example.*  *class Employee:*  *def \_\_init\_\_(self, first, last):*  *self.name=first*  *self.last=last*  *class Manager(Employee):*  *pass* |
| 13 | What is the difference between a superclass and a subclass? Provide an example *A subclass takes the attributes from a super(parent) class. A super() function is used to give access to the methods and attributes of the parent class. (not sure which the question is asking…)*  *class Employee: #Super/Parent Class*  *def \_\_init\_\_(self, first, last):*  *self.name=first*  *self.last=last*  *class Manager(Employee): #subclass*  *pass*  *super().\_\_init\_\_(first,last) #super function to inherit methods from Super/Parent*  *self.department = department* |
| 14 | When inheriting a class, why would you use **pass**? *Pass is used to inherit the attributes of a superclass into a subclass.* |
| 15 | Create a Pet subclass from an Animal superclass.  *class Animal: #Super/Parent Class*  *def \_\_init\_\_(self, color,fur):*  *self.color=color*  *self.fur=fur*  *class Pet(Animal): #subclass*  *pass* |
| **#** | **Chapter 12 – Creating a Graphical User Interface with Python** |
| 16 | What is the difference between a form and a widget? *A form is an object using built in classes provided by python, while a widget is an object on the form with which the user will interact with.* |
| 17 | What is **tkinter**? *Tkinter is a python library containing tools used to constructed basic graphical user interface applications in python.* |
| 18 | Create an empty form with a title “This is my form – Your name”  *from tkinter import\**  *frmFinal=Tk()*  *frmFinal.title("This is my form – Brian Sheehan")*  *frmFinal.mainloop()* |
| 19 | Create a form with a label showing today’s date.  *from tkinter import\**  *from datetime import date*  *frmFinal=Tk()*  *frmFinal.title("This is my form – Brian Sheehan")*  *today=date.today()*  *lblDate=Label(text=today, font="Arial 10")*  *lblDate.place(x=10,y=10)*  *frmFinal.mainloop()* |
| 20 | Create a form to sell sandwiches (i.e., small, medium, large) with meats (i.e., turkey, chicken, steak) at a sandwich shop with your name (i.e., Sam’s Best Sandwiches).  *See Sandwich Shop.py* |