|  |  |
| --- | --- |
| Course Name | ITD 2313 – Script Programming |
| Instructor | Put instruction name here |
| Student Name | Put student name here |
| Due date | Put assignment due date here |
| Grade | Put grade earned here |
| Grading Comments | Put instructor comments here |

# Accessing Characters and Substrings in Strings

## The Structure of Strings

### Page 88

1. The first code example block is at the top of the page is a simple review of some previous material dealing with strings.

A computer screen shot of a black screen

Description automatically generated

1. The 2nd code example block is about 1/2 way down the page starts looking into what we can do with strings.

A computer screen shot of a black screen

Description automatically generated

## The Subscript Operator

### Page 88

1. The first code example block begins at the bottom and does spill over onto the top of the next page this page is a series of commands.  There is even one that has an error in it.  Do not correct that error.  You need to see and capture that error in the screen shot.

A computer screen shot of a computer

Description automatically generated

1. The second code example block is about 1/2 way down the page.  The portion that spills over is the output so go ahead and put this screen shot with the page 88 grouping.

## Slicing for substrings

### Page 89

1. The code example block begins at the bottom and does spill over onto the top of the next page this page.  Slicing is an important concept when using strings in Python.  Be sure to do all of the code example block in the middle of the page.

A computer screen shot of a black screen

Description automatically generated

## Testing for a Substring with the in operator

### Page 90

1. There is a small code block example in the middle of the page to demonstrate how the operator works.

A screenshot of a computer

Description automatically generated

# Data Encryption

### Page 91-92

1. There are two rather large code example blocks here.  One to Encode and one to Decode.  The good news is that you do not have to type them in.  They are included in the data files for this chapter.  Run the programs multiple times using 3 at least one set of code and decode to show the versatility of the two programs.

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

# Strings and Number Systems

## Converting Binary to Decimal

### Page 94

1. The code example block at the bottom of the page.  This one is also available in your download for this chapter.  Run it and show the same inputs as the text and then run them again with different inputs to see what you get.

A computer screen shot of a black screen

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

## Converting Decimal to Binary

### Page 95

1. The code example block for this one is also available in your download for this chapter.  Run it and show the same inputs as the text and then run them again with different inputs to see what you get.

A computer screen shot of a black screen

Description automatically generated

# String Methods

### Page 97

1. The code example block is at the bottom of the page and does spill over onto the top of the next page.

A computer screen shot of a black screen

Description automatically generated

### Page 99

1. The first code example block on this page is rather long so it might take multiple screen shots to get this one and the next one on the page.

A screenshot of a computer

Description automatically generated

1. The second code example block on the page is at the bottom and does spill over onto the top of the next page. Again it might take multiple screen shots to get all the code example blocks on this page.

A computer screen shot of a black screen

Description automatically generated

# Text Files

## Writing Text to a File

### Page 101

1. This is an unusual one.  In this small section, there are several code lines but there is no output.  That is because the output is in a file.  So for this one, you will run the code lines start to finish and then grab the screen shot of all of them.  Then to show they worked, find the file it created on your hard drive and open it.  Then grab a screen shot of that file open in whatever text editor you use.

A screenshot of a computer

Description automatically generated

1. Do not delete the file when you are done, you will use it later.

## Reading Text from a file

### Page 101

1. Normally for a single page we would do all the code blocks together but the file handling is a bit special.  It uses a file you created earlier so make sure the file name is correct.  Nothing may show when you execute the first code example block in this subsection of the text.  Still screen shot that you did it.

A computer screen shot of a black screen

Description automatically generated

1. Executing the code example block just after the half way mark on the page will have output you can capture with your screen shot.

A computer screen shot of a black square

Description automatically generated

1. There will be output for the last code example block on this page to grab into the screen shot.

A computer screen shot of a black square

Description automatically generated

## Reading Numbers from a File

### Page 103

1. This code block at the top of your page.  It utilizes one of the files created earlier.  Make sure you have the correct file name.

A computer screen shot of a black screen

Description automatically generated

1. The code example block in the middle of the page uses that file again.  Check that file name to be sure.  There is output to capture for the screen shot.

## Accessing and Manipulating files and Directories on Disk

### Page 104

1. You can if you want do the code example blocks on this page.  Do not though include any screen shots into the submission document for this page.

### Page 106

1. This page does have a code example block you are to execute and run and grab that screen shot for the submission document.