School of Science, Computing and Artificial Intelligence
The University of the West Indies, Five Islands

COMP2232 - Assignment 0 1%

Introduction

This lab aims to reinforce the concepts of writing a program that only contains a main and writing a simple Java class.

Item	Marks
Exercise 1	10 marks
Exercise 2	30 marks
Exercise 3	50 marks

Lab exercises

1 Exercise 1: Fundamentals of Java Programming

Write a java file named JavaMCQ.java and do the following:

1. create two constant variables one for ”TRUE” and another for ”FALSE”.
2. print the following statements to the console and their corresponsing truth value, i.e. if the statement is true then use the variable to ouput the answer infront of the statement. for example. Java is an object-oriented language! TRUE etc. If the statement is false print the truth value (FALSE) add a single statement for the correct answer.
   (a) Reserved words in Java can be redefined by the programmer to mean something other than their original intentions.
   (b) In a Java program, dividing by zero is a syntax error.
   (c) Java byte codes are directly executable whereas Java source code is not.
   (d) The Java compiler is able to find all programmer errors.
   (e) Java is a case-sensitive language which means Current, CURRENT, and current will all reference the same identifier.

2 Exercise 2: Math Functions, Characters, Strings

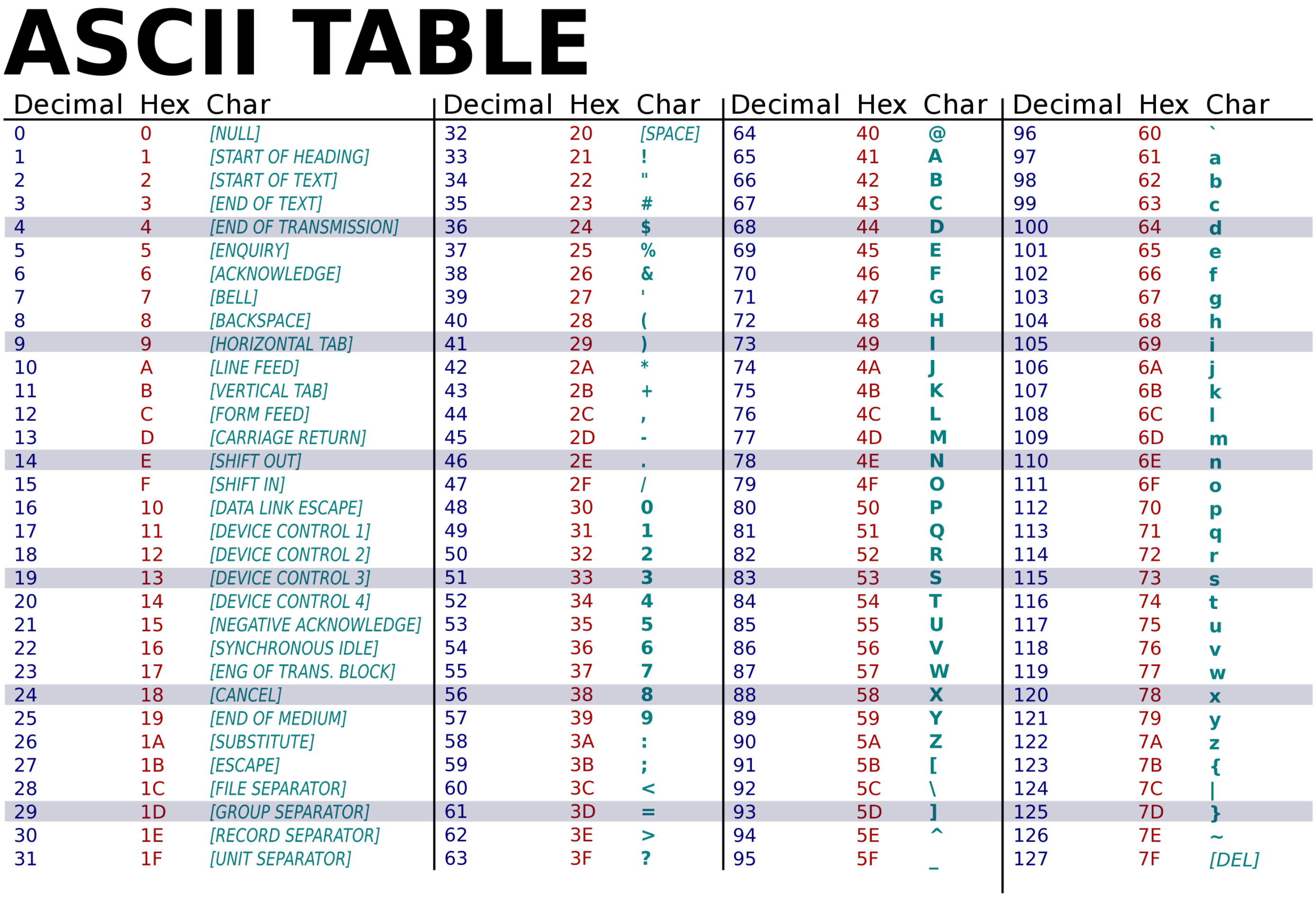
1. Consider the attached ASCII-table and do the following:
   (a) Write a program that receives an ASCII code (an integer between 0 and 127) and displays its character.
   (b) Add statemets that receives a character and displays its Unicode.
   (c) Write a program that prompts the user to enter a letter grade A, B, C, D, or F and displays its corresponding numeric value 4, 3, 2, 1, or 0.

2. Write a program that prompts the user to enter the x- and y-coordinates of the three corner points in a triangle then displays the three angles.

3. The great circle distance is the distance between two points on the surface of a sphere. Let (x1, y1) and (x2, y2) be the geographical latitude and longitude of two points. The great circle distance be- tween the two points can be computed using the following formula:

```
d = radius * arccos(sin (x1) * sin(x2) + cos(x1) * cos(x2) * cos(y1 - y2))
```

   Write a program that prompts the user to enter the latitude and longitude of two points on the earth in degrees and displays its great circle distance. The average radius of the earth is 6,371.01 km. Note you need to convert the degrees into radians using the Math.toRadians method since the Java trigonometric meth- ods use radians. The latitude and longitude degrees in the formula are for north and west. Use negative to indicate south and east degrees.



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

3 Exercise 3: Selections, Conditionals and Loops

1. The program in LoveCS.java prints “I love Computer Science!!” 10 times. Copy it to your directory and compile and run it to see how it works.

```
// *****************************************************************
//   LoveCS.java
//
//   Use a while loop to print many messages declaring your
//   passion for computer science
// *****************************************************************
public class LoveCS
{
```

```
public static void main(String[] args)
{
final int LIMIT = 10;
int count = 1;
 while (count <= LIMIT){
 }
System.out.println("I love Computer Science!!");
count++;
  }
 }
```

Then modify it as follows:

(a) Instead of using constant LIMIT, ask the user how many times the message should be printed. You will need to declare a variable to store the user's response and use that variable to control the loop. (Remember that all caps is used only for constants!)

(b) Number each line in the output, and add a message at the end of the loop that says how many times the message was printed. So if the user enters 3, your program should print this:

```
1 I love Computer Science!!
2 I love Computer Science!!
3 I love Computer Science!!
Printed this message 3 times.
```

(c) If the message is printed N times, compute and print the sum of the numbers from 1 to N. So for the example above, the last line would now read:

```
Printed this message 3 times. The sum of the numbers from  1 to 3 is 6.
```

Note that you will need to add a variable to hold the sum.

2. The factorial of n (written n!) is the product of the integers between 1 and n. Thus 4! = 1*2*3*4 = 24. By definition, 0! = 1. Factorial is not defined for negative numbers.

(a) Write a program that asks the user for a non-negative integer and computes and prints the factorial of that integer. You'll need a while loop to do most of the work—this is a lot like computing a sum, but it's a product instead. And you'll need to think about what should happen if the user enters 0.

(b) Now modify your program so that it checks to see if the user entered a negative number. If so, the program should print a message saying that a nonnegative number is required and ask the user the enter another number. The program should keep doing this until the user enters a nonnegative number, after which it should compute the factorial of that number. Hint: you will need another while loop before the loop that computes the factorial. You should not need to change any of the code that computes the factorial!

3. File Guess.java contains a skeleton for a program to play a guessing game with the user. The program should randomly generate an integer between 1 and 10, then ask the user to try to guess the number. If the user guesses incorrectly, the program should ask them to try again until the guess is correct; when the guess is correct, the program should print a congratulatory message.

(a) Using the comments as a guide, complete the program so that it plays the game as described above.

(b) Modify the program so that if the guess is wrong, the program says whether it is too high or too low. You will need an if statement (inside your loop) to do this.

(c) Now add code to count how many guesses it takes the user to get the number, and print this number at the end with the congratulatory message.

(d) Finally, count how many of the guesses are too high and how many are too low. Print these values, along with the total number of guesses, when the user finally guesses correctly.

```
// ************************************************************
//   Guess.java
//
//   Play a game where the user guesses a number from 1 to 10
//
// ************************************************************
import java.util.Scanner;
import java.util.Random;
public class Guess
{
public static void main(String[] args)
{
int numToGuess; //Number the user tries to guess int guess; //The user's guess
Scanner scan = new Scanner(System.in);
Random generator = new Random();
//randomly generate the  number to guess
//print message asking user to enter a guess
//read in guess
while (  )  //keep going as long as the guess is wrong
  {
//print message saying guess is wrong
//get another guess from the user
}
//print message saying guess is right
    }
}
```