Module 4:

Function Basics

COP2274

In-class Assignments

UF UNIVERSITY of FLORIDA

M4A Coin flip

1. Create a programmer-defined function called **coin_flip()** that simulates a coin flip. Each time the function **coin_filp()** is called, it should *randomly* return a character ‘H’ for heads or ‘T’ for tails.
2. Test **coin_flip()** in your main program as many as the user’s entered input. Your program should print out the result from each coin flip and the probability of getting heads (2 decimal places) as shown in the test cases.
3. Ask the user if they want to run the program again (y/n), repeat the program until the user enters 'n', and then exit with “Goodbye!” as shown in the test cases.

M4A Coin flip

Test cases

```
How many times do you like to toss your coin? 7
1: H
2: T
3: T
4: H
5: H
6: T
7: H
The probability of getting heads: 57.14%
Run it again (y/n)? y
How many times do you like to toss your coin? 4
1: T
2: T
3: H
4: T
The probability of getting heads: 25.00%
Run it again (y/n)? n
Goodbye!
```

Notes:

- *Your output will not be the same as the test cases due to randomness.*
- *You can include <cstdlib> and <ctime> for access to rand(), srand(), and time(). Seed the random number generator at the very beginning of the program by using srand(time(0)).*

M4B Rectangle with an alphabet

1. Create a programmer-defined function called **is_alphabet()** that takes in a character and returns “true” if the character is an alphabet (A-Z, a-z) or “false” if it is not an alphabet.
2. Create a programmer-defined function called **print_rectangle()** which takes in a length, height, and character and prints a rectangle made out of that character with the given length and height.
3. Test **is_alphabet()** and **print_rectangle()** in your main program as shown in the test cases.

M4B Rectangle with an alphabet

Notes:

- *Don't forget to ask the user if they want to print another rectangle.*
- *ASCII table*

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
64	40	100	@	@	96	60	140	`	`
65	41	101	A	A	97	61	141	a	a
66	42	102	B	B	98	62	142	b	b
67	43	103	C	C	99	63	143	c	c
68	44	104	D	D	100	64	144	d	d
69	45	105	E	E	101	65	145	e	e
70	46	106	F	F	102	66	146	f	f
71	47	107	G	G	103	67	147	g	g
72	48	110	H	H	104	68	150	h	h
73	49	111	I	I	105	69	151	i	i
74	4A	112	J	J	106	6A	152	j	j
75	4B	113	K	K	107	6B	153	k	k
76	4C	114	L	L	108	6C	154	l	l
77	4D	115	M	M	109	6D	155	m	m
78	4E	116	N	N	110	6E	156	n	n
79	4F	117	O	O	111	6F	157	o	o
80	50	120	P	P	112	70	160	p	p
81	51	121	Q	Q	113	71	161	q	q
82	52	122	R	R	114	72	162	r	r
83	53	123	S	S	115	73	163	s	s
84	54	124	T	T	116	74	164	t	t
85	55	125	U	U	117	75	165	u	u
86	56	126	V	V	118	76	166	v	v
87	57	127	W	W	119	77	167	w	w
88	58	130	X	X	120	78	170	x	x
89	59	131	Y	Y	121	79	171	y	y
90	5A	132	Z	Z	122	7A	172	z	z
91	5B	133	[	[	123	7B	173	{	{
92	5C	134	\	\	124	7C	174			|
93	5D	135	]	]	125	7D	175	}	}
94	5E	136	^	^	126	7E	176	~	~
95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Test cases

```
Enter an alphabetical character: 2
Enter an alphabetical character, either A-Z or a-z: c
Enter length and height, separated by a space: 7 4
ccccccc
c     c
c     c
ccccccc
Would you like to print another rectangle? (y/n) y
Enter an alphabetical character: p
Enter length and height, separated by a space: 4 7
pppp
p  p
p  p
p  p
p  p
p  p
pppp
Would you like to print another rectangle? (y/n) n
Goodbye!
```

M4C Dice rolling guessing game

1. Create a programmer-defined function called **roll()** that returns a random integer between 1 and 6 inclusive.
2. Create a programmer-defined function called **start_game()** that that takes in a double for the amount of money a user has, and returns the amount of money they have after playing the game. Each game costs $1.50 to play. Take the following steps to define your **start_game()**.
   - Prompt the user to guess an integer between 1-6 with input validation. Assume that the user always enters an integer.

M4C Dice rolling guessing game

- Roll a dice by calling your **roll()** function and display the number that was rolled as shown in the test cases.
- If the user guesses the correct number, add $2.00 to user money and display "Your guess was correct! You win $2.00." If the user guesses an incorrect number, display "Your guess was incorrect! You lose." as shown in the test cases.
- Return the updated user money.

3. Take the following steps in your main program.
   - User will start the game with $5.00. Print how much money they have at the start of the program.

M4C Dice rolling guessing game

- Ask the user with input validation if they would like to play for $1.50 (y/n)?
- If they would like to play, start the game by calling your **start_game()** function, that takes in the amount of money they currently have, and returns the new amount after playing the game. If they would not, end your program with a goodbye message.
- Your program should keep playing game as long as user has more than or equal to $1.50. If they do not have enough money, display the message and end your program with a goodbye message.

M4C Dice rolling guessing game

Test case1

```
Welcome to the dice rolling guessing game!

You currently have: $5.00
Would you like to play for $1.50? (y/n): n

Goodbye!
```

Test case 2

```
Welcome to the dice rolling guessing game!

You currently have: $5.00
Would you like to play for $1.50? (y/n): u
Invalid entry!
Would you like to play for $1.50? (y/n): y
Guess a positive number (1-6): 8
Invalid entry!
Guess a positive number (1-6): 4
You rolled a: 4
Your guess was correct! You win $2.00!
```

```
You currently have: $5.50
Would you like to play for $1.50? (y/n): y
Guess a positive number (1-6): 5
You rolled a: 3
Your guess was incorrect! You lose.

You currently have: $4.00
Would you like to play for $1.50? (y/n): y
Guess a positive number (1-6): 3
You rolled a: 4
Your guess was incorrect! You lose.

You currently have: $2.50
Would you like to play for $1.50? (y/n): y
Guess a positive number (1-6): 1
You rolled a: 1
Your guess was correct! You win $2.00!

You currently have: $3.00
Would you like to play for $1.50? (y/n): y
Guess a positive number (1-6): 3
You rolled a: 6
Your guess was incorrect! You lose.
```

M4C Dice rolling guessing game

Test case 2 (contd.)

```
You currently have: $1.50
Would you like to play for $1.50? (y/n): y
Guess a positive number (1-6): 4
You rolled a: 6
Your guess was incorrect! You lose.
Sorry, you do not have enough money!

Goodbye!
```

Notes:

- *Your output will not be the same as the test cases due to randomness.*
- *You can include <cstdlib> and <ctime> for access to rand(), srand(), and time(). Seed the random number generator at the very beginning of the program by using srand(time(0)).*