Unit 03
Programming Fundamentals

Programming Activities

Activity 1

In the context of fig, 3.10, rewrite code to swap the values of 2 variables without using temporary variable.

Program

```
x = 5
y = 3
print("Values of x: ", x, " and y:", y)
Print("Lets swap values of x and y")
x, y = y, x
print("Values of x: ", x, " and y= ", y)
```

Activity 2

Take three inputs as day, month and year e.g. your date of birth, store in three variables and print in the form of:

- **13-09-2023**
- **09-13-2023**
- **2023-09-13**

Program

```
#input values
day=input ("Enter day (dd):")
month=input ("Enter month (mm):")
year=input ("Enter year (yyyy):")
#Print dates
Print(day + "-" + month + "-" + year)
Print(month + "-" + day + "-" + year)
Print(year + "-" + month + "-" + day)
```

Activity 3

Take input from the user which determines the height of the diamond and prints a hollow diamond (◇).

Program

```
# Set the number of rows for the top half of the diamond
```

federalpastpapers.com

```
h = int(input('Enter height of diamond:'))
# Top half of the diamond
for i in range(h):
    for j in range(h - i - 1):
        print(" ", end="")
    for j in range(2 * i + 1):
        if j == 0 or j == 2 * i:
            print("*", end="")
        else:
            print(" ", end="")
    print()

# Bottom half of the diamond
for i in range(h-2, -1, -1):
    for j in range(h - i - 1):
        print(" ", end="")
    for j in range(2 * i + 1):
        if j == 0 or j == 2 * i:
            print("*", end="")
        else:
            print(" ", end="")
    print()
```

Activity 4

Take a random 2-digit number. Subtract it from 100. Calculate the difference. Print the difference (magnitude only).

Program

```
import random
#Generate a random two-digit number
    number=random.randint(10,99)
#calculate the difference
     diff=100-number
#print magnitude of difference
     Print("Random 2-digit number:", number)
     Print("Magnitude of difference:", abs(diff))
```

Activity 5

Take a number as input from the user, using a function, print a mathematical table.

federalpastpapers.com

Program

```
def table():
      number= eval(input("Enter a table number:"))
      For i in range(1,11):
            Print(number,"x", i, "=", number*i)
#calling table function
table()
```

Activity 6

Take input from the user which determines the height of the diamond and print a filled diamond (◆).

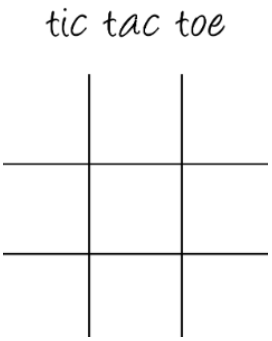
Program

```
def print_filled_diamond(height):
# Ensure height is an odd number
    if height % 2 == 0:
        height += 1
    mid = height // 2
    # Top half of the diamond
    for i in range(mid + 1):
        print(' ' * (mid - i) + '◆' * (2 * i + 1))
    # Bottom half of the diamond
    for i in range(mid - 1, -1, -1):
        print(' ' * (mid - i) + '◆' * (2 * i + 1))
# Input from user
height = int(input("Enter the height of the diamond: "))
print_filled_diamond(height)
```

Activity 7

Write a turtle graphics code to draw a Tic-Tac-Toe board.

Program




```
import turtle
def draw_line(x1, y1, x2, y2):
    turtle.penup()
    turtle.goto(x1, y1)
    turtle.pendown()
    turtle.goto(x2, y2)

def draw_tic_tac_toe_board():
    turtle.speed(3)
```

federalpastpapers.com

```
    turtle.pensize(3)

# Draw the vertical lines
    draw_line(-100, 100, -100, -100)
    draw_line(100, 100, 100, -100)

# Draw the horizontal lines
    draw_line(-100, 100, 100, 100)
    draw_line(-100, -100, 100, -100)
    turtle.hideturtle()
    turtle.done()

# Run the function to draw the Tic-Tac-Toe board
draw_tic_tac_toe_board()
```

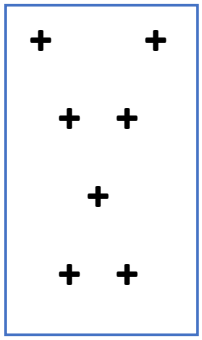
Activity 8

Take an odd number as input from the user and draw a 'X' with the help of a character like '+'. The input number defines the size of 'X' to be displayed.

Program

```
def draw_x(size):
    if size % 2 == 0:
        print("Please enter an odd number.")
        return
    for i in range(size):
        for j in range(size):
            if i == j or i + j == size - 1:
                print('+', end='')
            else:
                print(' ', end='')
        print()

# Input from user
size = int(input("Enter an odd number for the size of 'X': "))
draw_x(size)
```



Activity 9

Write a program to calculate how many iterations it takes to subtract a small number from a larger one. Where both the numbers are positive integrals.

Program

```
def count_subtractions(larger, smaller):
    if smaller <= 0:
        raise ValueError("The smaller number must be a positive integer.")
    if larger < smaller:
```

Federalpastpapers.com

```
        raise ValueError("The larger number must be greater than or equal
to the smaller number.")
       iterations = 0
    while larger > 0:
        larger -= smaller
        iterations += 1
        return iterations
# Example usage
larger_number = 20
smaller_number = 3
iterations = count_subtractions(larger_number, smaller_number)
print(f"It takes {iterations} iterations to subtract {smaller_number} from
{larger_number} until the result is less than or equal to zero.")
```

federalpastpapers.com