Programming Assignment-14

Question 1:

Ans.

```
class DivisibleBySeven:
    def __init__(self, n):
        self.n = n

    def generator(self):
        for num in range(0, self.n + 1):
            if num % 7 == 0:
                yield num

# Example
n = int(input("Enter the upper limit: "))
div_by_seven = DivisibleBySeven(n)
for number in div_by_seven.generator():
    print(number)
```

Question 2:

Ans.

```
def word_frequency(sentence):
    words = sentence.split()
    freq_dict = {}
    for word in words:
        freq_dict[word] = freq_dict.get(word, 0) + 1
    sorted_freq = sorted(freq_dict.items())

    for word, freq in sorted_freq:
        print(f"{word}:{freq}")
```

```
# Example input
sentence = input("Enter a sentence: ")
word_frequency(sentence)
```

Example:

Input: `New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.`

Output:

2:2

3.:1

3?:1

New:1

Python:5

Read:1

and:1

between:1

choosing:1

or:2

to:1

Question 3:

Ans.

```
class Person:
    def getGender(self):
        pass

class Male(Person):
    def getGender(self):
```

```
        return "Male"

class Female(Person):
    def getGender(self):
        return "Female"

# Example
male = Male()
female = Female()

print(male.getGender())  # Output: Male
print(female.getGender())  # Output: Female
```

Question 4:

Ans.

```
def generate_sentences(subjects, verbs, objects):
    for subject in subjects:
        for verb in verbs:
            for obj in objects:
                print(f"{subject} {verb} {obj}.")

# Example
subjects = ["I", "You"]
verbs = ["Play", "Love"]
objects = ["Hockey", "Football"]
generate_sentences(subjects, verbs, objects)
```

Output:

```
I Play Hockey.
```

I Play Football.

I Love Hockey.

I Love Football.

You Play Hockey.

You Play Football.

You Love Hockey.

You Love Football.

Question 5:

Ans.

```
import zlib

def compress_string(input_string):
    compressed = zlib.compress(input_string.encode())
    return compressed

def decompress_string(compressed_string):
    decompressed = zlib.decompress(compressed_string).decode()
    return decompressed

# Example
input_string = "hello world!hello world!hello world!hello world!"
compressed_str = compress_string(input_string)
print("Compressed string:", compressed_str)

decompressed_str = decompress_string(compressed_str)
print("Decompressed string:", decompressed_str)
```

Question 6:

Ans.

```
def binary_search(arr, target):
    low, high = 0, len(arr) - 1

    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1

    return -1  # If not found

# Example
arr = [1, 3, 5, 7, 9, 11]
target = int(input("Enter the number to search: "))
index = binary_search(arr, target)
if index != -1:
    print(f"Element found at index {index}")
else:
    print("Element not found")
```