Assignment 5:
Student ID: 901142
Student Name: Roshan Shrestha

Source code:

```python
def check_month(entered_month):
    included_months = ['january', 'february', 'march', 'april', 'may', 'june',
                       'july', 'august', 'september', 'october', 'november', 'december']
    return entered_month.lower() in included_months

def check_day(entered_day):
    return 1 <= entered_day <= 31

def parse_date(entered_date):
    splitted_string = entered_date.strip().split(' ')
    if len(splitted_string) != 2:
        return None
    return [splitted_string[0].lower(), int(splitted_string[1])]

  # Determine the season based on the entered date
def get_season(month, day):
    if (month == 'march' and day >= 20) or (month == 'april') or (month == 'may') or (month == 'june' and day < 21):
        return 'Spring'
    elif (month == 'june' and day >= 21) or (month == 'july') or  (month == 'august') or (month == 'september' and day < 22):
        return 'Summer'
    elif (month == 'september' and day >= 22) or  (month == 'october') or (month == 'november') or (month == 'december' and day < 21):
        return 'Fall'
    else:
        return 'Winter'

def main():
    # Ask the user to input the date in the format: [month day]
    entered_date = input("Please enter a date e.g July 10 : ")

    # Parse the entered date to extract month and day
    parsed_date = parse_date(entered_date)
    if parsed_date is None:
        print("Please enter in the correct format as 'January 20.")
        return

    entered_month = parsed_date[0].lower()
    entered_day = parsed_date[1]

    # Check if the parsed month and day is valid
    if not check_month(entered_month):
        print(
            "OOPS !, you entered invalid month.\nMonth must be between January to December.")
        return

    if not check_day(entered_day):
        print("OOPS !, you entered invalid day.\nDay must be between 1 to 31.")
        return

    # Determine and get the season based on entered month and day
    season = get_season(entered_month, entered_day)
    print(season)

main()
```

Test case 1 [March 20]:

```python
def check_month(entered_month):
    included_months = ['january', 'february', 'march', 'april', 'may', 'june',
                       'july', 'august', 'september', 'october', 'november', 'december']
    return entered_month.lower() in included_months


def check_day(entered_day):
    return 1 <= entered_day <= 31


def parse_date(entered_date):
    splitted_string = entered_date.strip().split(' ')
    if len(splitted_string) != 2:
        return None
    return [splitted_string[0].lower(), int(splitted_string[1])]

    # Determine the season based on the entered date


def get_season(month, day):
    if (month == 'march' and day >= 20) or (month == 'april') or (month == 'may') or (month == 'june' and day < 21):
        return 'Spring'
    elif (month == 'june' and day >= 21) or (month == 'july') or (month == 'august') or (month == 'september' and day < 22):
        return 'Summer'
    elif (month == 'september' and day >= 22) or (month == 'october') or (month == 'november') or (month == 'december' and day < 21):
        return 'Fall'
    else:
        return 'Winter'


def main():
    # Ask the user to input the date in the format: [month day]
    entered_date = input("Please enter a date e.g July 10 : ")

    # Parse the entered date to extract month and day
    parsed_date = parse_date(entered_date)
    if parsed date is None:
```

COMMENTS   PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**

```
> python3 assign5.py
Please enter a date e.g July 10 : march 20
Spring
```

Test case 1 [July 15]:

```python
def check_month(entered_month):
    included_months = ['january', 'february', 'march', 'april', 'may', 'june',
                       'july', 'august', 'september', 'october', 'november', 'december']
    return entered_month.lower() in included_months


def check_day(entered_day):
    return 1 <= entered_day <= 31


def parse_date(entered_date):
    splitted_string = entered_date.strip().split(' ')
    if len(splitted_string) != 2:
        return None
    return [splitted_string[0].lower(), int(splitted_string[1])]

    # Determine the season based on the entered date


def get_season(month, day):
    if (month == 'march' and day >= 20) or (month == 'april') or (month == 'may') or (month == 'june' and day < 21):
        return 'Spring'
    elif (month == 'june' and day >= 21) or (month == 'july') or (month == 'august') or (month == 'september' and day < 22):
        return 'Summer'
    elif (month == 'september' and day >= 22) or (month == 'october') or (month == 'november') or (month == 'december' and day < 21):
        return 'Fall'
    else:
        return 'Winter'


def main():
    # Ask the user to input the date in the format: [month day]
    entered_date = input("Please enter a date e.g July 10 : ")

    # Parse the entered date to extract month and day
    parsed_date = parse_date(entered_date)
    if parsed_date is None:
```

```
> python3 assign5.py
Please enter a date e.g July 10 : july 15
Summer
```

Test case 1 [November 15]:

```python
def check_month(entered_month):
    included_months = ['january', 'february', 'march', 'april', 'may', 'june',
                       'july', 'august', 'september', 'october', 'november', 'december']
    return entered_month.lower() in included_months


def check_day(entered_day):
    return 1 <= entered_day <= 31


def parse_date(entered_date):
    splitted_string = entered_date.strip().split(' ')
    if len(splitted_string) != 2:
        return None
    return [splitted_string[0].lower(), int(splitted_string[1])]

    # Determine the season based on the entered date


def get_season(month, day):
    if (month == 'march' and day >= 20) or (month == 'april') or (month == 'may') or (month == 'june' and day < 21):
        return 'Spring'
    elif (month == 'june' and day >= 21) or (month == 'july') or (month == 'august') or (month == 'september' and day < 22):
        return 'Summer'
    elif (month == 'september' and day >= 22) or (month == 'october') or (month == 'november') or (month == 'december' and day < 21):
        return 'Fall'
    else:
        return 'Winter'


def main():
    # Ask the user to input the date in the format: [month day]
    entered_date = input("Please enter a date e.g July 10 : ")

    # Parse the entered date to extract month and day
    parsed_date = parse_date(entered_date)
    if parsed_date is None:
```

COMMENTS   PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**

```
› python3 assign5.py
Please enter a date e.g July 10 : November 15
Fall
```

Test case 1 [sep]:

```python
def check_month(entered_month):
    included_months = ['january', 'february', 'march', 'april', 'may', 'june',
                       'july', 'august', 'september', 'october', 'november', 'december']
    return entered_month.lower() in included_months


def check_day(entered_day):
    return 1 <= entered_day <= 31


def parse_date(entered_date):
    splitted_string = entered_date.strip().split(' ')
    if len(splitted_string) != 2:
        return None
    return [splitted_string[0].lower(), int(splitted_string[1])]

    # Determine the season based on the entered date


def get_season(month, day):
    if (month == 'march' and day >= 20) or (month == 'april') or (month == 'may') or (month == 'june' and day < 21):
        return 'Spring'
    elif (month == 'june' and day >= 21) or (month == 'july') or (month == 'august') or (month == 'september' and day < 22):
        return 'Summer'
    elif (month == 'september' and day >= 22) or (month == 'october') or (month == 'november') or (month == 'december' and day < 21):
        return 'Fall'
    else:
        return 'Winter'


def main():
    # Ask the user to input the date in the format: [month day]
    entered_date = input("Please enter a date e.g July 10 : ")

    # Parse the entered date to extract month and day
    parsed_date = parse_date(entered_date)
    if parsed_date is None:
```

```
> python3 assign5.py
Please enter a date e.g July 10 : sep
Please enter in the correct format as 'January 20.
```