Python Basics Assignment

Modules 1-6: Basics to Functions

CampusPe

Total Questions: 20 | Maximum Marks: 100 | Passing Marks: 60

Assignment Instructions

1. Test your code with multiple inputs before submission.

2. Add comments explaining your logic throughout your code.

3. Use meaningful and descriptive variable names.

4. Handle edge cases and invalid inputs using try-except where appropriate.

5. Include a README.txt with your name, questions attempted, special instructions, and challenges faced.

Marks Distribution

Q#	Topic	Difficulty	Marks
Q1	Personal Bio Card	Easy	2
Q2	Simple Calculator	Easy	2
Q3	String Manipulator	Medium	4
Q4	Age Calculator	Medium	4
Q5	Bill Splitter	Medium	4
Q6	Grade Calculator	Easy-Medium	4
Q7	Temperature Converter	Medium	4
Q8	Leap Year Checker	Medium	4
Q9	Ticket Pricing System	Medium	4
Q10	Simple ATM Simulator	Medium-Hard	7
Q11	Number Pattern Printer	Medium	4
Q12	Multiplication Table Generator	Easy-Medium	4
Q13	Sum and Average Calculator	Easy	4
Q14	Factorial Calculator	Medium	4
Q15	Prime Number Checker	Medium-Hard	7
Q16	Number Guessing Game	Hard	7
Q17	Palindrome Checker	Medium	4
Q18	Calculator Functions	Medium	7
Q19	Text Analysis Functions	Hard	9
Q20	Number System Functions	Hard	9
		Total	**100**

Grading Rubric

Code Functionality (60%): Correct Output (30%), Edge Cases (15%), Efficiency (15%)

Code Quality (25%): Readability (10%), Comments (8%), Variable Names (7%)

Problem Understanding (15%): Requirements Met (10%), Completeness (5%)

Important Notes

Academic Integrity: Write your own code. Copying results in zero marks.

Resources Allowed: Official Python documentation, course materials, Stack Overflow (syntax reference only).

Not Allowed: Copy-pasting code, using AI tools to write code, sharing complete solutions.

Question 1: Personal Bio Card

Difficulty: Easy | Points: 2 | Time: 10 min

Write a program that displays your information in a formatted card:

```
╔════════════════════════════════╗
║       STUDENT BIO CARD         ║
╔════════════════════════════════╗
║ Name    : John Doe             ║
║ Age     : 20 years             ║
║ Course  : Python Programming   ║
║ College : ABC University       ║
║ Email   : john@example.com     ║
╚════════════════════════════════╝
```

Requirements:

- Use variables for each field
- Display in a nicely formatted box
- Must be visually appealing

Question 2: Simple Calculator

Difficulty: Easy | Points: 2 | Time: 15 min

Create a program that:

1. Asks user for two numbers
2. Performs and displays: Addition, Subtraction, Multiplication, Division, Modulus, Exponentiation

Sample Output:

```
Enter first number: 10
Enter second number: 3
Results:
10 + 3 = 13
10 - 3 = 7
10 * 3 = 30
10 / 3 = 3.33
10 % 3 = 1
10 ^ 3 = 1000
```

Question 3: String Manipulator

Difficulty: Medium | Points: 4 | Time: 20 min

Ask user for a sentence and display:

1. Original sentence
2. Total characters (with spaces)
3. Total characters (without spaces)
4. Total words
5. UPPERCASE
6. lowercase
7. Title Case
8. First word
9. Last word
10. Reversed sentence

Sample Output:

```
Enter a sentence: Hello World Python
Original: Hello World Python
Characters (with spaces): 18
Characters (without spaces): 16
Words: 3
UPPERCASE: HELLO WORLD PYTHON
lowercase: hello world python
Title Case: Hello World Python
First word: Hello
Last word: Python
Reversed: nohtyP dlroW olleH
```

Question 4: Age Calculator

Difficulty: Medium | Points: 4 | Time: 20 min

Ask user for their birth year and calculate:

1. Current age
2. Age in months
3. Age in days (approx 365 days/year)
4. Age in hours
5. Age in minutes
6. Years until age 100

Bonus (+2 points): Ask for an exact birth date (day, month, year) and calculate more precisely.

Question 5: Bill Splitter

Difficulty: Medium | Points: 4 | Time: 25 min

Create a restaurant bill splitting program.

Inputs: Total bill amount, Number of people, Tax percentage, Tip percentage

Calculate and Display: Subtotal, Tax amount, Bill after tax, Tip amount, Total bill, Amount per person

Sample Output:

```
Enter total bill: 1000
Number of people: 4
Tax percentage: 10
Tip percentage: 15

=== BILL BREAKDOWN ===
Subtotal:    ₹1000.00
Tax (10%):   ₹100.00
After tax:   ₹1100.00
Tip (15%):   ₹165.00
Total:       ₹1265.00
Per person:  ₹316.25
```

Question 6: Grade Calculator

Difficulty: Easy-Medium | Points: 4 | Time: 20 min

Ask users for marks in 5 subjects (out of 100 each). Calculate and display:

1. Marks in each subject
2. Total marks (out of 500)
3. Percentage
4. Grade
5. Result: Pass/Fail (Pass if all subjects >= 40)

Grade Scale:

90-100% : A+ (Outstanding)
80-89% : A (Excellent)
70-79% : B (Good)
60-69% : C (Average)
50-59%: D (Pass)
Below 50%: F (Fail)

Question 7: Temperature Converter

Difficulty: Medium | Points: 4 | Time: 25 min

Create a temperature converter with a menu-based system supporting:

1. Celsius to Fahrenheit

2. Fahrenheit to Celsius

3. Celsius to Kelvin

4. Kelvin to Celsius

5. Fahrenheit to Kelvin

6. Kelvin to Fahrenheit

7. Exit

Formulas:

- C to F: (C × 9/5) + 32

- F to C: (F - 32) × 5/9

- C to K: C + 273.15

- K to C: K - 273.15

- F to K: (F - 32) × 5/9 + 273.15

- K to F: (K - 273.15) × 9/5 + 32

Question 8: Leap Year Checker

Difficulty: Medium | Points: 4 | Time: 20 min

Create a program that checks if a year is a leap year.

Rules: A year is a leap year if divisible by 4 AND (NOT divisible by 100 OR divisible by 400).

Test Cases:

2024: Leap year

2100: NOT leap year

2000: Leap year

2023: NOT leap year

Display the year, whether it is a leap year, and the reason why.

Question 9: Ticket Pricing System

Difficulty: Medium | Points: 4 | Time: 25 min

Create a movie ticket pricing system.

Age-based Pricing:

Below 3: Free

3-12: ₹150 (Child)

13-59: ₹300 (Adult)

60+: ₹200 (Senior)

Day-based Discount:

Monday-Thursday: No discount

Friday-Sunday: 20% discount

Inputs: Age, Day of week, Number of tickets

Display base price, discount (if any), price after discount, and total amount.

Question 10: Simple ATM Simulator

Difficulty: Medium-Hard | Points: 7 | Time: 30 min

Create an ATM simulation with initial balance = ₹10,000.

Menu: 1. Check Balance 2. Deposit Money 3. Withdraw Money 4. Exit

Rules:

- Check sufficient balance before withdrawal

- Minimum balance of ₹500 must remain at all times

- Display transaction messages and updated balance after each transaction

Sample Run:

```
ATM SIMULATOR
    1. Check Balance
    2. Deposit
    3. Withdraw
    4. Exit
    Enter choice: 3
    Enter amount to withdraw: 2000
    Withdrawal successful!
    New balance: ₹8000
```

Question 11: Number Pattern Printer

Difficulty: Medium | Points: 4 | Time: 25 min

Create a program that prints the following patterns. User should choose which pattern and height.

Pattern 1:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Pattern 2:

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Pattern 3:

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Pattern 4:

```
    1
   121
  12321
 1234321
123454321
```

Question 12: Multiplication Table Generator

Difficulty: Easy-Medium | Points: 4 | Time: 20 min

Create a program that asks the user for a number and a range, then displays the multiplication table.

Sample Output:

```
Enter number: 7

Enter range (end): 12

Multiplication Table of 7

7 x 1 = 7

7 x 2 = 14

...

7 x 12 = 84
```

Bonus (+3 points): Create a full multiplication table (1-10 for all numbers 1-10) in grid format.

Question 13: Sum and Average Calculator

Difficulty: Easy | Points: 4 | Time: 20 min

Ask the user how many numbers they want to add. Then take that many numbers as input using a loop.

Calculate: 1. Sum 2. Average 3. Maximum number 4. Minimum number

Sample Output:

```
How many numbers? 5
Enter number 1: 10
Enter number 2: 20
...
Sum: 100
Average: 20.0
Maximum: 30
Minimum: 10
```

Question 14: Factorial Calculator

Difficulty: Medium | Points: 4 | Time: 20 min

Calculate factorial of a number using a loop. Factorial: $n! = n \times (n-1) \times (n-2) \times \dots \times 1$

Requirements:

- Handle 0 and negative numbers

- Display step-by-step calculation

Sample Output:

```
Enter a number: 5
5! = 5 × 4 × 3 × 2 × 1 = 120
```

Question 15: Prime Number Checker

Difficulty: Medium-Hard | Points: 7 | Time: 30 min

Part 1 (5 marks): Check if a single number is prime. Handle negative numbers, 0, 1, and 2.

Part 2 (2 marks): Find all prime numbers in a given range.

Sample Output:

```
Enter a number: 17
17 is a PRIME number

Enter start range: 1
Enter end range: 30
Prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29
```

Question 16: Number Guessing Game

Difficulty: Hard | Points: 7 | Time: 35 min

Create a number guessing game where the computer picks a random number between 1-100 and the user gets 7 attempts.

Rules:

1. After each guess, show if guess is too high or too low and attempts remaining

2. If correct: congratulate and show attempts used

3. If failed: reveal the number

4. Ask to play again

Bonus (+3 points): Track best score (minimum attempts) and give hints when close (within 5).

Hint: Use import random and random.randint(1, 100)

Question 17: Palindrome Checker

Difficulty: Medium | Points: 4 | Time: 20 min

Create a program that checks if a word/number is a palindrome (reads same forwards and backwards).

Requirements: Check words (ignore case), check numbers, display step-by-step verification.

Sample Output:

```
Enter word/number: Racecar
Original: Racecar
Reversed: racecaR
Result: PALINDROME
```

Question 18: Calculator Functions

Difficulty: Medium | Points: 7 | Time: 30 min

Create a calculator using functions.

Required Functions:

1. add(a, b)

2. subtract(a, b)

3. multiply(a, b)

4. divide(a, b) - handle division by zero

5. modulus(a, b)

6. power(a, b)

7. calculator() - main function with menu

The calculator() function should display a menu, take user input, call the appropriate function, and display the result. Include an Exit option.

Question 19: Text Analysis Functions

Difficulty: Hard | Points: 9 | Time: 40 min

Create the following text analysis functions:

1. count_words(text) - return number of words

2. count_vowels(text) - return number of vowels

3. count_consonants(text) - return number of consonants

4. reverse_text(text) - return reversed text

5. is_palindrome(text) - return True/False

6. remove_vowels(text) - return text without vowels

7. word_frequency(text) - return dictionary of word counts

8. longest_word(text) - return longest word

9. analyze_text(text) - calls all above functions and displays results

Sample Output:

```
Enter text: Hello World Hello
=== TEXT ANALYSIS ===
Words: 3
Vowels: 4
Consonants: 8
Reversed: olleH dlroW olleH
Palindrome: No
Without vowels: Hll Wrld Hll
Longest word: Hello (5 letters)
Word Frequency: hello: 2, world: 1
```

Question 20: Number System Functions

Difficulty: Hard | Points: 9 | Time: 45 min

Create the following mathematical functions:

1. factorial(n) - return n!

2. is_prime(n) - return True if prime

3. fibonacci(n) - return nth Fibonacci number

4. sum_of_digits(n) - return sum of digits

5. reverse_number(n) - return number reversed

6. is_armstrong(n) - check if Armstrong number (e.g., $153 = 1^3 + 5^3 + 3^3$)

7. gcd(a, b) - greatest common divisor

8. lcm(a, b) - least common multiple

9. is_perfect_number(n) - sum of divisors equals n (e.g., 6 = 1+2+3)

10. math_menu() - menu to test all functions

Each function should be callable individually from the menu with appropriate user input.

Tips

1. Start with easy questions to build confidence.
2. Test thoroughly with different inputs.
3. Handle errors using try-except for user input.
4. Read each question carefully and understand all requirements.
5. Comment your code to explain complex logic.
6. Ask for help if stuck for more than 30 minutes.
7. Manage your time and don't spend too long on one question.

Bonus Opportunities

Complete any 3 bonus challenges for extra points:

1. Enhanced ATM (Q10): Add transaction history
2. Pattern Generator (Q11): Add 3 more creative patterns
3. Game Features (Q16): Add difficulty levels (Easy: 1-50, Medium: 1-100, Hard: 1-1000)
4. Advanced Calculator (Q18): Add square root, percentage functions
5. Statistical Analysis (Q13): Add median, mode calculations

Good Luck!

Focus on learning, not just completing. Understanding the concepts is more important than finishing all the questions.