

Review Test Submission: Midterm Test Part 2 (Debugging)

User	Arzu Gizem Kirici
Course	Introduction to Object Oriented Programming
Test	Midterm Test Part 2 (Debugging)
Started	10/17/22 2:35 PM
Submitted	10/17/22 3:05 PM LATE
Due Date	10/17/22 3:05 PM
Status	Completed
Attempt Score	18 out of 40 points
Time Elapsed	30 minutes out of 30 minutes
Results Displayed	Feedback

Question 1

10 out of 10 points

The program is expected to calculate the area of Circle and Square by taking in radius of the Circle and side of the rectangles as parameters through parameterized constructors of each class. There is a global method called `void calculateArea(Square square, Circle circle)` which is responsible for calculating and displaying the area. Expected output of the program is:

```
Area of Circle : 50.24
Area of Square : 25
```

However, currently, the program has some logical and compile time errors. Please fix them to make the program work as expected.

```
#include<iostream>
using namespace std;

class Circle
{
private:
    int radius;

    Circle(int r)
    {
        radius = this->r;
    }
};

class Square
{
private:
    int side;

    Square(int s)
    {
        s = s;
    }
};

void calculateArea(Square square, Circle circle)
{
    float circle_area, square_area;

    circle_area = 3.14 * ((circle.r) * (circle.r));
    cout << "Area of Circle : " << circle_area << endl;

    square_area = square.s * square.s;
    cout << "Area of Square : " << square_area << endl;
}
```

```

int main()
{
    Circle circle = Circle('4');
    Square square = Square('5');

    calculateArea(square, circle);
    return 0;
}

```

Response Feedback: [None Given]

Question 2

8 out of 10 points

The Program below is expected to take a date in the form of day, month and year from the user and displays it as **Date: 17-8-2006**. Currently, the program has some logical and compile time errors, please fix the program so that it can work as expected. Do not change the parameter names in the constructor.

```

#include <iostream>
using namespace std;

class Date
{
private:
    int day, month, year;

public:
    Date(int day, int month, int year)
    {
        day = day;

        if (-100 <= month <= 100)
        {
            month = month;
        }
        else
        {
            month = '1';
        }

        year = year;
    }

    void displayDate()
    {
        cout << "Date: "<<day << "-" << month << "-" << year << endl;
    }
};

int main()
{
    Date date = Date(17, 8);
    Date.displayDate();
    return 0;
}

```

Response Feedback: if (-100 <= month && month <= 100) should be if (1 <= month <= 12)
 this->month = '1'; should be this->month = 1;

Question 3

0 out of 10 points

The program below is expected to store the even numbers into the heap memory. It asks user about how many numbers to be entered in an array. Once user enters the size of the array, it asks user to enter some numbers. Based on the entry of the user, the program checks if the numbers are even or not. If the numbers are even, it counts the even numbers and then create an array of integers into the heap

memory. For example, if user enters 5 as array size and enters 1, 2, 3, 4, 5 as numbers, an array of size 2 is created in the heap memory as there are only 2 even numbers from the collection user provided. The program should then display the even numbers stored in the heap memory. Below is the expected output of the program:

```
Enter the number of elements to store in an array: 5
Enter a number to store in array: 1
Enter a number to store in array: 2
Enter a number to store in array: 3
Enter a number to store in array: 4
Enter a number to store in array: 5
```

```
Array of Even numbers...
2      4
```

Currently program has some missing lines of code. Fill in those missing lines to make the program work as expected.

```
#include <iostream>
using namespace std;

int EVEN_SIZE = 0;
void display(int* arr, int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << "\t";
    }
}

void isEven(int *arr, int size)
{
    for (int i = 0; i < size; i++)
    {
        if (// missing condition)
        {
            EVEN_SIZE++;
        }
    }

    // missing line of code for creating heap memory
    int j = 0;
    for (int i = 0; i < size; i++)
    {
        if (arr[i] % 2 == 0)
        {
            // missing line of code to enter elements from original
            // array to array of even numbers.
            j++;
        }
    }
    cout << "\nArray of Even numbers..." << endl;
    display(ptr, EVEN_SIZE);
}

int main()
{
    int size, number;
    cout << "Enter the number of elements to store in an array: ";
    cin >> size;
    int* ptr = new int[size]{ 0 };

    for (int i = 0; i < size; i++)
    {
        cout << "Enter a number to store in array: ";
        cin >> number;
```

```

        //missing line of code to enter element to array
    }

    isEven(ptr, size);

    return 0;
}

```

Response Feedback: [None Given]

Question 4

0 out of 10 points

The program below is expected to keep taking numbers from the user until user presses -1. The program is expected to calculate the sum of all the numbers and stores it in a sum variable in the heap throughout all the iterations of the loop. Once user presses -1, the sum should be shown along with its address in the memory. Below is the expected output of the program:

```

Enter a number (-1 to quit): 1
Enter a number (-1 to quit): 2
Enter a number (-1 to quit): 3
Enter a number (-1 to quit): 4
Enter a number (-1 to quit): 5
Enter a number (-1 to quit): -1
Sum in heap: 15
Address of Sum in heap: 00C80FB8

```

Some of the lines of the program are missing. You are required to figure out the missing lines and complete the code.

```

#include <iostream>
using namespace std;

int main()
{
    int number;
    // missing line of code
    while (true)
    {
        cout << "Enter a number (-1 to quit): ";
        cin >> number;

        if (// missing if condition)
        {
            break;
        }

        // missing line for calculating sum from heap
    }

    cout << "Sum in heap: " << *ptrSum << endl;
    cout << "Address of Sum in heap: " << ptrSum << endl;

    // deallocating the sum variable in heap
    // fixing the dangling pointer
    return 0;
}

```

Response Feedback: [None Given]