**PROGARM 5**

QUESTION:

Create a class Twin Prime having following specification.

Class Name – TwinPrime

Data Methods:

int n1 – to store the first number.

int n2 – to store the second number.

Member Methods:

TwinPrime (int nn, int mm) – to initialize n1 with nn and n2 with mm.

int Prime (int n) – to check whether the number is prime or not. If its prime return 1 else 0.

void check() – check whether n1 and n2 is twin prime or not by calling prime(int n) function.

Write a main() to create the object of the class and call the above methods properly.

ALGORITHM:

1. Start
2. Define a class named `TwinPrime`.
3. Declare two integer instance variables `n1` and `n2` to

store the input numbers.

1. Create a constructor `TwinPrime(int nn, int mm)` to initialize `n1` and `n2` with the given values.

Algorithm for the `Prime` Method:

1. Start with a parameter `n` (the number to be checked for primality).
2. If `n` is less than or equal to 1, return 0 (indicating not prime).
3. Iterate `i` from 2 to the square root of `n`:
   * Check if `n` is divisible evenly by `i` (i.e., `n % i == 0`).
     + If divisible, return 0 (indicating not prime).
4. If no divisors were found, return 1 (indicating prime).

Algorithm for the `check` Method:

1. Call the `Prime` method with `n1` as an argument and store the result in a variable `prime1`.
2. Call the `Prime` method with `n2` as an argument and store the result in a variable `prime2`.

1. Calculate the absolute difference between `n1` and `n2` and store it in a variable `absDiff`.
2. If both `prime1` and `prime2` are equal to 1 (indicating both numbers are prime) and `absDiff` is equal to 2, then:
   * Print "They are twin prime."
3. Otherwise, print "They are not twin prime."

Algorithm for the `main` Method:

1. Initialize a `Scanner` object named `sc` to read input from the user.
2. Print the message: "Enter first number."
3. Read an integer from the user and store it in a variable named `a`.
4. Print the message: "Enter second number."
5. Read another integer from the user and store it in a variable named `b`.
6. Create an instance of the `TwinPrime` class named `number`, passing `a` and `b` as arguments to the constructor.
7. Call the `check()` method on the `number` object to determine if the numbers are twin primes.
8. End the `main` method.

**VARIABLE DESCRIPTION TABLE**

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Variable Description |
| nn | int | An integer |
|  |  | representing the first |
|  |  | input number used |
|  |  | to initialize `n1` in |
|  |  | the `TwinPrime` |
|  |  | constructor. |
| mm | int | An integer |
|  |  | representing the |
|  |  | second input number |
|  |  | used to initialize `n2` |
|  |  | in the `TwinPrime` |
|  |  | constructor. |
| n1 | int | An integer |
|  |  | representing the first |
|  |  | number to be |
|  |  | checked for twin |
|  |  | prime status. It is |
|  |  | initialized with the |
|  |  | value of `nn` in the |
|  |  | `TwinPrime` |
|  |  | constructor. |
| n2 | int | An integer |
|  |  | representing the |
|  |  | second number to be |
|  |  | checked for twin |
|  |  | prime status. It is |
|  |  | initialized with the |
|  |  | value of `mm` in the |
|  |  | `TwinPrime` |
|  |  | constructor. |
| i | int | A loop variable |
|  |  | used in the `Prime` |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | method to iterate | |
|  |  | through numbers | |
|  |  | from 2 to the square | |
|  |  | root of the input | |
|  |  | number (`n`) while | |
|  |  | checking for | |
|  |  | primality. | |
| a | int | An integer | |
|  |  | variable that holds | |
|  |  | the user's input for | |
|  |  | the first number. It is | |
|  |  | read using a | |
|  |  | `Scanner` object in | |
|  |  | the `main` method. | |
| b | Int | An integer | |
|  |  | variable that holds | |
|  |  | the user's input for | |
|  |  | the second number. | |
|  |  | It is read using a | |
|  |  | `Scanner` object in | |
|  |  | the `main` method. | |
| sc | Scanner | A `Scanner` object | |
|  |  | used to read input | |
|  |  | from the user in the | |
|  |  | `main` method. | |
| number | TwinPrime | An instance of the |  |
|  |  | `TwinPrime` class |  |
|  |  | created using the |  |
|  |  | input numbers `a` |  |
|  |  | and `b`. It is used to |  |
|  |  | call the `check()` |  |
|  |  | method to |  |
|  |  | determine if the |  |
|  |  | numbers are twin |  |
|  |  | primes. |  |