

# Core Java

## assignments

---

### Assignment #1

#### Prime or not?

Implement the body for the following function:

```
static boolean isPrimeNumber(int num) {  
    // do stuff here  
    return false;  
}
```

The function should check and return true only if the number passed as argument is a prime number.

Write a Java program to call the above function multiple times with different values.

---

### Assignment #2

#### Fibonacci

In mathematics, the Fibonacci numbers are the numbers in the following integer sequence, characterised by the fact that every number after the first two numbers is the sum of the two preceding ones:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Write a Java function called "fibonacci", that takes index as parameter and returns the fibonacci number at that index.

```
public static int fibonacci(int index){  
    // do stuff here  
    return 0;  
}
```

Test the working of the same by calling it multiple times with different index values in the `main()` method of your class.

---

# Assignment #3

## Reverse a sentence by words

Write a function called "reverseByWords", that takes a sentence (string) as an input, and returns another string. The return value must be a sentence in which the words in the original sentence appear in reverse order.

```
public String reverseByWords(String sentence) {  
    // do stuff here  
    return null;  
}
```

For example,

```
String out = reverseByWords("my name is vinod and i live in bangalore");  
// the variable "out" should be equal to "bangalore in live i and vinod is name my".
```

Call the function in main, multiple times by supplying multiple values and verify the same.

---

# Assignment #4

## Calendar for given month and year

Implement the Java function listed below:

```
public static void printCalendar(int month, int year) {  
    /// do stuff here  
}
```

The function should accept **month** and **year** and print the calendar for the same. If inputs are invalid, appropriate error message/s should be printed.

Sample output for the inputs (8, 2018):

```
Su Mo Tu We Th Fr Sa  
      1  2  3  4  
 5  6  7  8  9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30 31
```

PS:

- Do not use any builtin Java classes like **Date** or **Calendar**
  - Divide the function into small reusable functions, if possible.
-

# Assignment #5

## Number to words

Write a function called "inWords" that takes a number between 1 and 99,99,99,999 and returns a String representing the input number in words.

```
public String inWords(int num) {  
    // do stuff here  
    return null;  
}
```

For example,

```
inWords(12345);  
// should return "twelve thousand three hundred forty five"  
inWords(10203040);  
// should return "one crore two lakh three thousand forty"  
inWords(101);  
// should return "one hundred one"
```

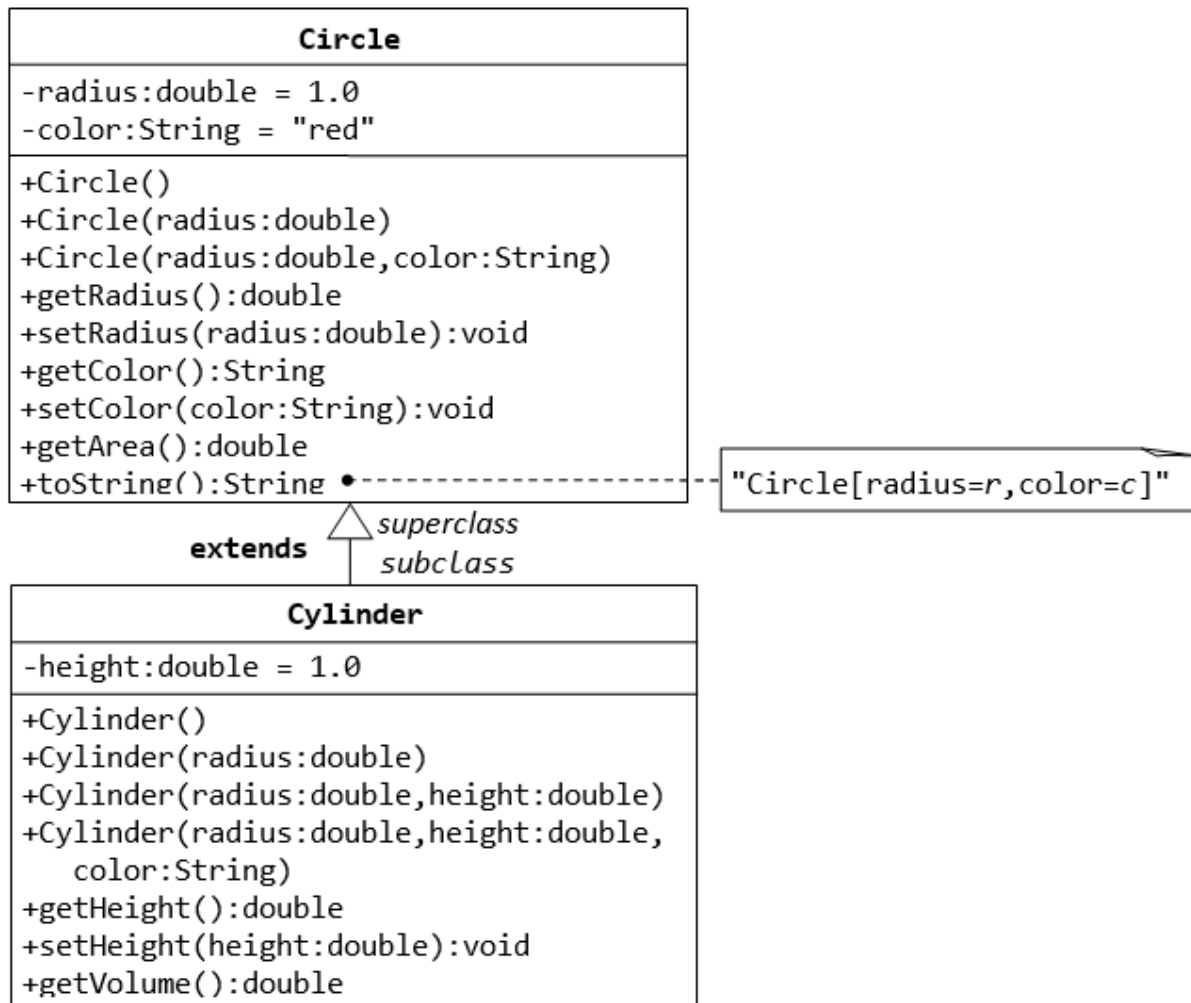
Call the function in main, multiple times by supplying multiple values and verify the same.

---

# Assignment #6

## Inheritance and polymorphism

Create classes **Circle** and **Cylinder** as shown in the *UML* diagram below:



In the `main()` function of a Program class, create an array of **Circle** references with the initialization shown below:

```
Circle[] circles = {
    new Cylinder(12.34),
    new Cylinder(12.34, 10.0),
    new Cylinder(12.34, 10.0, "blue")
};
```

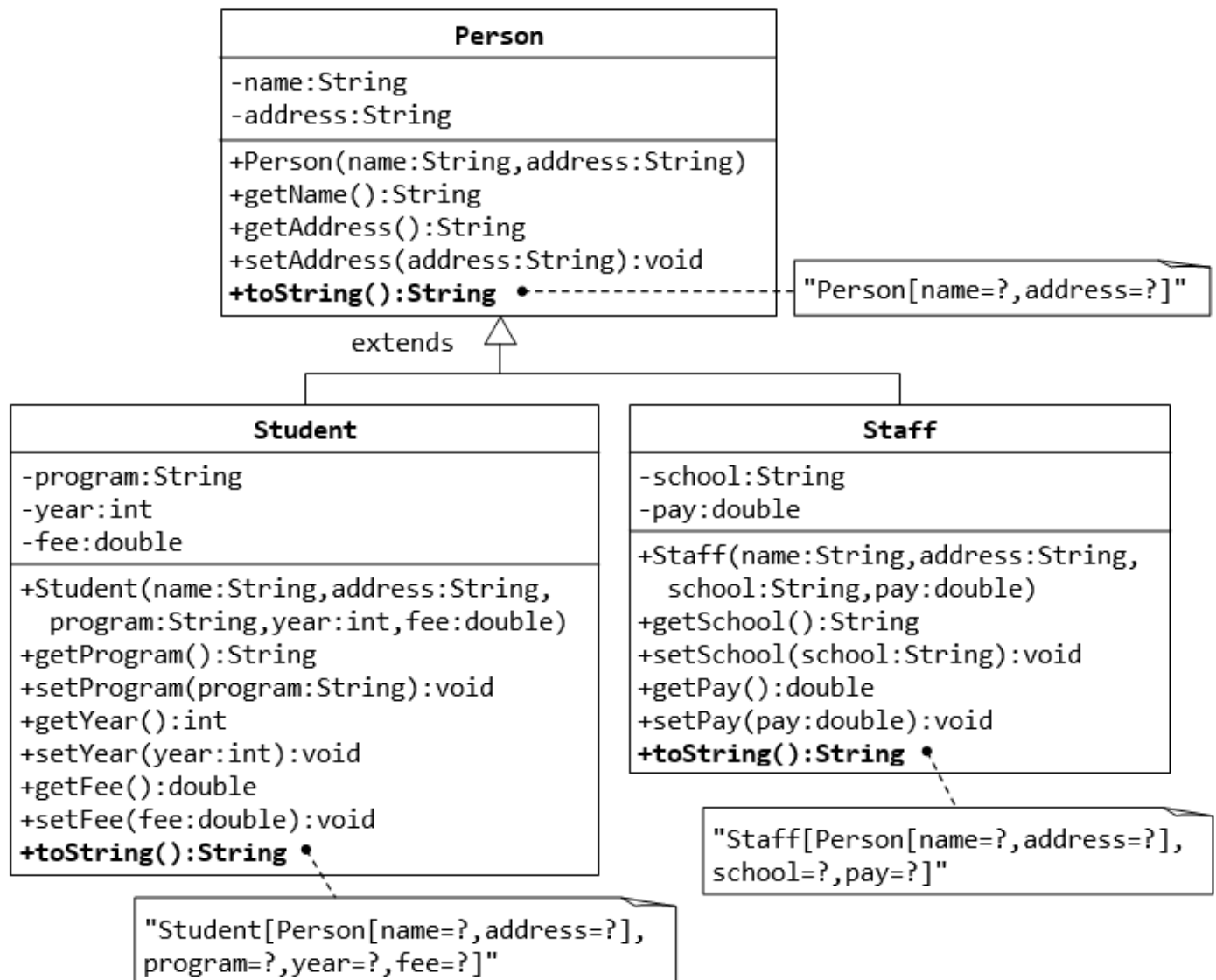
Print the area of the circular region of each cylinder along with the volume of the same.

---

# Assignment #7

## Classes, inheritance and polymorphism

Create the classes **Person**, **Student**, and **Staff** as shown in the *UML* diagram below:



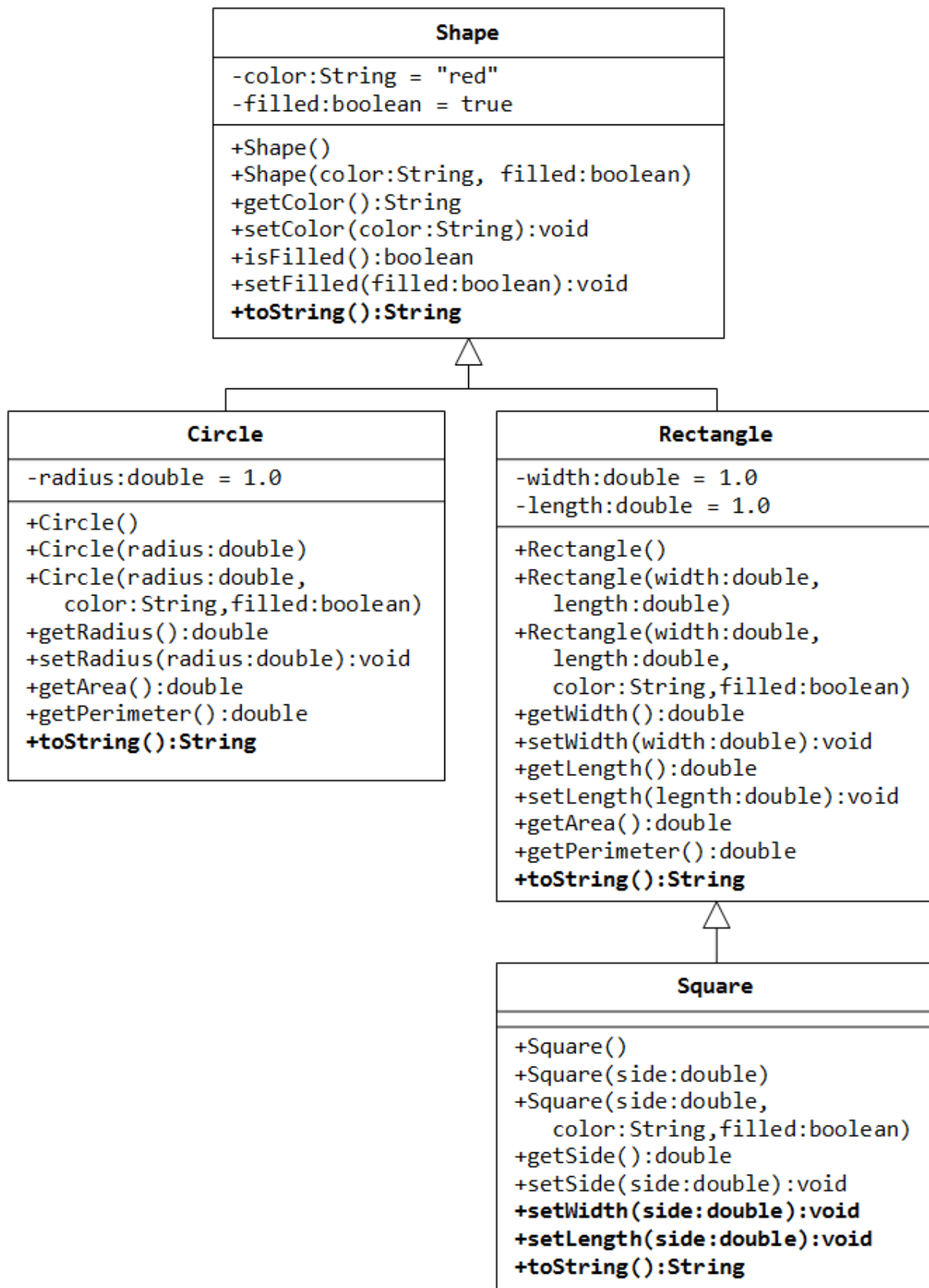
In the `main()` function of a `Program` class, create an array of **Person** references with the initialisation shown below:

```
Person[] people = {
    new Student("Shyam", "Bangalore, Karnataka", "Java fundamentals", 2010, 4500.0),
    new Staff("Anand", "Bangalore, Karnataka", "Delhi Public school", 35000.0),
    new Staff("Umesh", "Bangalore, Karnataka", "National Public school", 42000.0),
    new Student("Suresh", "Hassan, Karnataka", "Java fundamentals", 2012, 4750.0),
    new Student("Kiran", "Vasco, Goa", "ReactJS", 2017, 12500.0)
};
```

Print the details of all **Person** objects (using the `toString()`).

## Classes, inheritance and polymorphism

Create the classes **Shape**, **Circle**, **Rectangle**, and **Square** as shown in the UML diagram below:



The **toString** function of the above classes should return text as given below:

| Classname | Sample return value from toString() | --- | --- | | Shape | A Shape with color of xxx and filled/Not filled | | Circle | A Circle with radius=xxx, which is a subclass of yyy (where yyy is the output of the toString() method from the superclass) | | Rectangle | A Rectangle with width=xxx and length=zzz, which is a subclass of yyy (where yyy is the output of the toString() method from the superclass) | | Square | A Square with side=xxx, which is a subclass of yyy (where yyy is the output of the toString() method from the superclass) |

In the `main()` method of a Program class, create an array of 10 `Shape` references containing a mixture of `Circle`, `Rectangle` and `Square` objects of different dimensions. Using a loop, print the `perimeter` and `area` for all of them.