

# SE1011: Exam 2 Name:

---

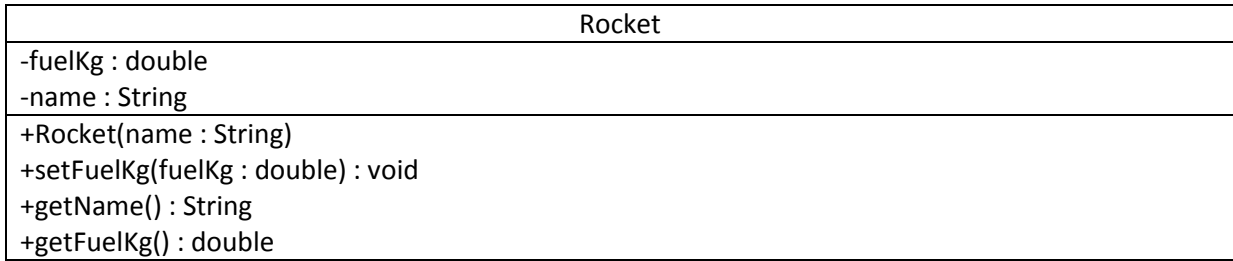
This is a closed-book, closed-computer, etc. exam. You may use one 8.5"x11" sheet of notes, which you will turn in with your exam. Review all questions before you get started. Use the page at the end of the exam for extra work. The exam is printed double-sided. Show all work.

1. (15 pts.) Given a `String str`, write a sequence of Java statements to print every other character in the string. Use the console (`System.out`).

```
String str = in.next();
```

2. (20pts.) Short answer problems. Be clear & concise.
- a. (4 pts.) In Java, how do you encapsulate an instance variable?
  
  
  
  
  
  
  
  
  
  
  - b. (4 pts.) If you do not write a constructor, how do you get an instance of the class?
  
  
  
  
  
  
  
  
  
  
  - c. (4 pts.) What is the difference between a class and an object?
  
  
  
  
  
  
  
  
  
  
  - d. (4 pts.) What does it mean to declare a method “private”? How does this affect the code?
  
  
  
  
  
  
  
  
  
  
  - e. (4 pts.) How can you tell if a variable is a local variable or an instance variable?

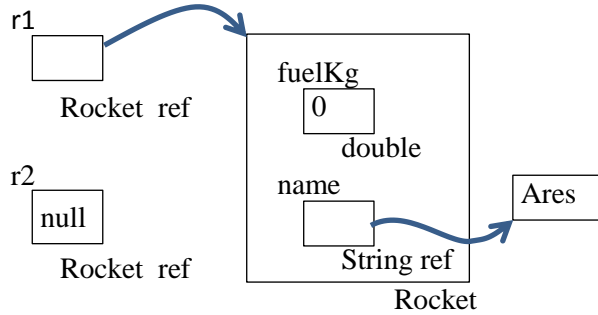
3. (20 pts.) Write a complete implementation of the class shown in the UML diagram below. You do not need to comment your methods



Further space for problem 3.

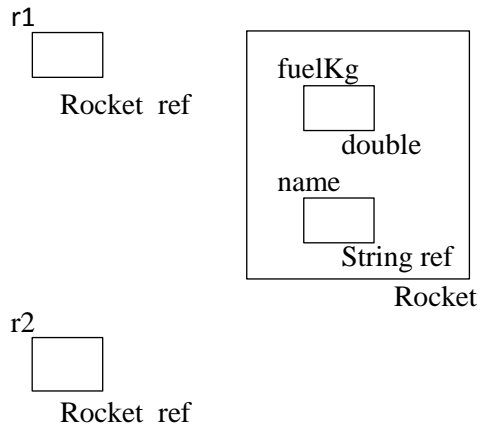
4. (20 pts.) Complete the following diagram that illustrates the state of memory at four points in a program on the right. Step a has been filled in entirely as an example.

//a



```
Rocket r1=null;
Rocket r2=null;
r1 = new Rocket("Ares");
// a
r1.setFuelKg(500);
r2 = r1;
// b
r1 = new Rocket("Saturn V");
// c
```

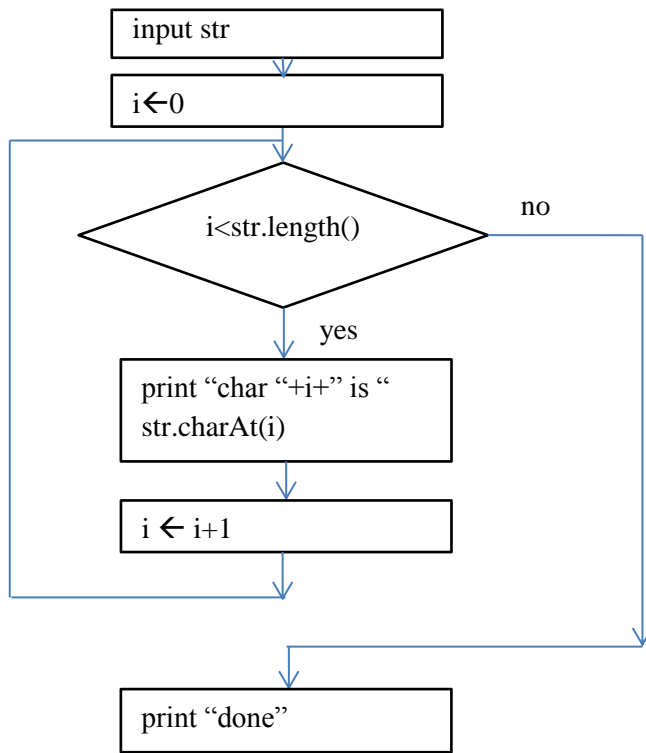
//b [COMPLETE THIS]



//c [COMPLETE THIS]



6. (10 pts.) Convert the following flow-chart to low-level pseudo-code



Pseudo-code:

You may use this page for extra work. Please write on the original problem page that you are using this space, and write here the original problem number.



# Partial Java API

---

Some of these methods should be used with an object, some should be used with the class name. Some methods show example arguments instead of the parameter definitions.

```
double Double.parseDouble(String str)
int Integer.parseInt(String str)
void JOptionPane.showInputDialog(null,String str)
void JOptionPane.showMessageDialog(null, String str)
long Long.parseLong(String str)
double Math.abs(double a)
double Math.acos(double a)
double Math.asin(double a)
double Math.atan(double a)
double Math.atan2(double a, double b)
double Math.cos(double a)
double Math.log(double a)
double Math.log10(double a)
double Math.max(double a, double b)
double Math.min(double a, double b)
double Math.pow(double a, double b)
double Math.PI
double Math.sqrt(double a)
double Math.sin(double a)
double Math.tan(double a)
void System.out.print()
void System.out.println()
Scanner.Scanner(System.in)
int Scanner.nextInt();
String Scanner.next();
String Scanner.nextLine();
char String.charAt(int pos);
int String.length();
```