

SYSC 2004 Assignment #1 Marking Scheme

The assignment is marked out of 10. Part marks are available, unless otherwise indicated. Round the final mark up to the nearest 0.5 mark, e.g. 6.75 would be 7, etc.

-1 (**deduction**): If anything other than exactly 5 files (CommandWords.java, Game.java, Item.java, Parser.java, Room.java) were submitted.

1: **CommandWords.java**, broken down as:

0.5: four new commands added to the validCommands field.

0.5: void showAll() changed to String getCommandList(), and String correctly returned, including the Javadoc update

-0.5 (**deduction**): if student has not updated Javadoc at top of class

0.5: **Parser.java**, broken down as:

0.5: void showCommands() changed to String getCommands() and String correctly returned, including 0.5 for the Javadoc update

-0.5 (**deduction**): if student has not updated Javadoc at top of class

2: **Item.java**, broken down as:

0.5: class and class Javadoc

0.5: fields

0.5: constructor, including Javadoc

0.5: getDescription(), including Javadoc

2: **Room.java**, broken down as:

0.25: ArrayList field added

0.25: constructor updated to initialize ArrayList

0.5: addItem() method, including Javadoc: this method may pass in the description and weight, instead of an Item object

0.5: getLongDescription() updated, including Javadoc (explaining that items now included)

0.5: getItems() method, including Javadoc: optional method – this functionality may be included in the getLongDescription() method, in which case it is worth 1 mark.

-0.5 (**deduction**): if student has not updated Javadoc at top of class

4.5: **Game.java**, broken down as:

0.5: fields previousRoom (for back) and Stack<Room> (for stackBack) added

0.5: constructor updated

0.5: createRooms() updated to create items and put them in rooms, including 0.25 marks for having at least one item in each room

0.5: processCommand() updated for the four new commands

0.5: goRoom() updated to deal with back and stackBack

0.5: look() method added, including Javadoc

0.5: eat() method added, including Javadoc

0.5: back() method added, including Javadoc

0.5: stackBack() method added, including Javadoc

-0.5 (**deduction**): if student has not updated Javadoc at top of class

Additional deductions (0.5 to 2 marks) should be made if code is significantly less efficient / more complicated than required, as code should demonstrate a good understanding of using other classes and use of internal and external method calls.

1 mark deduction if name and student number are both missing from any class Javadoc comments.

0.5 mark deduction if one of name or student number is missing from any class Javadoc comments.

Additional methods may be added but they must be **private**.

Late submissions are only accepted with permission of the instructor and there may be a deduction associated with that. (In those cases, I will notify you via email.)