1.1 What are the basic tasks that all software engineering projects must handle?

Development, testing, low-level design, high-level design, requirements, maintenance, deployment, wrap-up

1.2 Give a one sentence description of each of the tasks listed in Exercise 1.

Development: The act of producing the necessary code.
Testing: Testing the application's various features under a multitude of conditions to make sure it functions properly in all foreseeable scenarios.
Low level design: Having a detailed understanding about how each major piece of the project should function.
High level design: Understanding the major ideas of of the project, knowing what platforms are to be used, and knowing which database structure is to be used.
Requirements: Understanding what the customer wants/needs.
Maintenance: Monitoring and addressing any issues with the product.
Deployment: Distribute the product to users.
Wrap-up: Evaluate the project and decide what went well and what didn't.

2.5 What does JBGE stand for and what does it mean?

"Just Barely Good Enough"
Commenting code in such a way that it is not overly cumbersome to constantly update it as code is changed but thorough enough to explain the code to someone who has not personally worked on it.

3.2 What are the tasks on the critical path? What is the total expected duration of the project in working days?

Critical path: G, D, E, M, Q
Duration: 32 days

3.4 See Chart

3.6 How can you handle these sorts of unpredictable programs?

One can add the new tasks to the schedule and factor in the amount of time lost when reevaluating it.

3.8 What are the two biggest mistakes you can make while tracking tasks?

1. Assigning more people onto a task believing that it will reduce the ammount of time that task will take when in reality it does the opposite.
2. If a task is not going as planned, one must take action. Not doing so could lead to further problems.

4.1 List five characteristics of good requirements.

1. Easily understood
2. Unambiguous
3. Consistent
4. Prioritized
5. Verifiable

4.3

There are no implementation requirements
a. Business
b. Functional, user
c. Functional, user
d. Functional, user
e. Nonfunctional

f . Nonfunctional
g. Nonfunctional
h. Nonfunctional
i. Nonfunctional
j. Functional
k. Functional
m. Functional, user
n. Functional, user
o. Functional, user
p. Functional, user

4.9

Add a high score chart, so one can compare his or her attempt to others..
Add difficulty choices for different levels of play.