Player Character: How could you design your system so that each race could be easily generated? Additionally, how diﬃcult does such a solution make adding additional races?

The system could be designed so there is one player class, which has a series of subclasses for each race, connected by inheritance. If implemented correctly, subclasses would easily be made to add additional races.

Enemies: How does your system handle generating diﬀerent enemies? Is it diﬀerent from how you generate the player character? Why or why not?

The enemies and the character would be the same design, both of which would be under a main character class.

Enemies: How could you implement the various abilities for the enemy characters? Do you use the same techniques as for the player character races? Explain.

Potions: The Decorator and Strategy 8 patterns are possible candidates to model the eﬀects of potions, so that we do not need to explicitly track which potions the player character has consumed on any particular ﬂoor. In your opinion, which pattern would work better? Explain in detail, by weighing the advantages/disadvantages of the two patterns.

Treasure: How could you generate items so that the generation of Treasure and Potions reuses as much code as possible? That is, how would you structure your system so that the generation of a potion and then generation of treasure does not duplicate code?