

2008 AP Computer Science A Exam

1) "Trip" Class

(a)

```
public int getDuration() {  
  
    int s = flights.size();  
    if (s==0) return s;  
  
    Time first = flights.get(0).getDepartureTime();  
    Time last = flights.get(s-1).getArrivalTime();  
  
    return first.minutesUntil(last);  
  
}
```

(b)

```
public int getShortestLayover() {  
  
    int s = flights.size();  
    if (s<=1) return -1;  
  
    Time t1 = flights.get(0).getDepartureTime();  
    Time t2 = flights.get(1).getArrivalTime();  
  
    int shortest = t1.minutesUntil(t2);  
  
    for(int x = 1; x < s-1; x++){  
  
        Time first = flights.get(x).getDepartureTime();  
        Time next = flights.get(x+1).getArrivalTime();  
  
        int minutes = first.minutesUntil(next);  
  
        if (minutes <= shortest) shortest = minutes;  
  
    }  
  
    return shortest;  
  
}
```

2) “StringCoder” Class

(a)

```
public String decodeString(ArrayList<StringPart> parts) {  
  
    String result = "";  
  
    for(int x = 0; x<parts.size(); x++) {  
  
        int len = parts.get(x).getLength();  
        int start = parts.get(x).getStart();  
  
        result = result + masterString.substring(start, start+len);  
  
    }  
    return result;  
  
}
```

(b)

```
public ArrayList<StringPart> encodeString(String word){  
  
    ArrayList<StringPart> result = new ArrayList<StringPart>;  
  
    int x = 0;  
  
    while(x<word.length()){  
  
        String searchWord = word.substring(x);  
        StringPart sp = findPart(searchWord);  
        result.add(sp);  
        x = x + sp.getLength();  
    }  
  
    return result;  
  
}
```

3) “OpossumCriticr” Class

(a)

```
public void processActors (ArrayList<Actor> actors) {  
  
    int friendcount = 0;  
    int foecount = 0;  
  
    for(int x = 0; x<actors.size(); x++) {  
        if (isFriend(actors.get(x))) friendcount++;  
        else if (isFoe(actors.get(x))) foecount++;  
  
    }  
  
    if (foecount>friendcount) {  
        setColor(Color.BLACK);  
        numStepsDead++;  
  
    } else {  
        setColor(Color.ORANGE);  
        numStepsDead=0;  
    }  
}
```

(b)

```
public Location selectMoveLocation (ArrayList<Location> locs) {  
  
    Location loc = new Location();  
  
    if (getColor().equals(Color.BLACK)) {  
  
        if (numStepsDead>=3) loc = null; else loc = getLocation();  
  
    } else {  
  
        loc = super.selectMoveLocation(locs);  
  
    }  
  
    return loc;  
}
```

4) “Checker” Class

(a)

```
public class SubStringChecker implements Checker {  
    private String substr;  
  
    public SubStringChecker(String tp) {  
        substr = tp;  
    }  
  
    public boolean accept(String text) {  
        if (text.indexOf(substr)==-1) return false; else return true;  
    }  
}
```

(b)

```
public class AndChecker implements Checker{  
    private Checker chk1;  
    private Checker chk2;  
  
    public AndChecker(Checker tp1, Checker tp2){  
        chk1 = tp1.;  
        chk2 = tp2;  
    }  
  
    public boolean accept(String text){  
        if (chk1.accept(text) && chk2.accept(text)) return true; else return false;  
    }  
}
```

(c)

```
yummyChecker = AndChecker (NotChecker (aChecker) ,NotChecker (kChecker) );
```