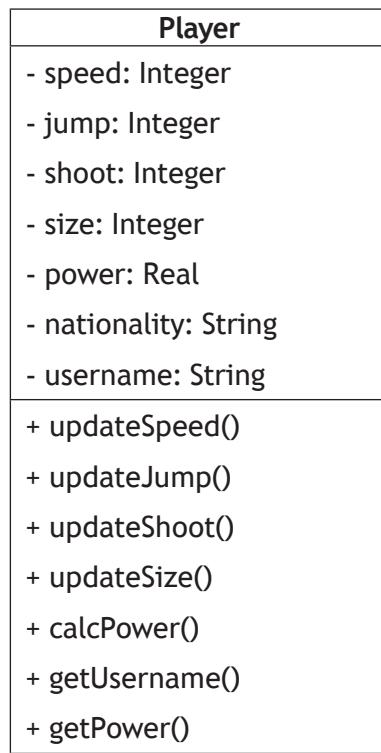


4. A game is being developed where the player controls a footballer, playing against an online opponent.



When the game is played for the first time, a player must provide a nationality and a username. A new player starts with 1 point for their speed, jump, shoot, size and power characteristics. By winning games, players can gain extra points in each of these characteristics.

A simplified UML class diagram for a `Player` object is shown below.



Some of the class declaration code is shown on the page opposite.

**4. (continued)**

```
CLASS Player IS { INTEGER speed, INTEGER jump, INTEGER shoot,
INTEGER size, INTEGER real, STRING nationality , STRING
username }
```

METHODS

```
CONSTRUCTOR(STRING nationality , STRING username)
    DECLARE THIS.speed INITIALLY 1
    DECLARE THIS.jump INITIALLY 1
    DECLARE THIS.shoot INITIALLY 1
    DECLARE THIS.size INITIALLY 1
    DECLARE THIS.power INITIALLY 1
    DECLARE THIS.nationality INITIALLY nationality
    DECLARE THIS.username INITIALLY username
END CONSTRUCTOR

PROCEDURE updateSpeed()
    SET THIS.speed TO THIS.speed + 1
END PROCEDURE

FUNCTION getUsername() RETURNS STRING
    RETURN THIS.username
END FUNCTION

FUNCTION getPower() RETURNS REAL
    RETURN THIS.power
END FUNCTION

...
END CLASS
```

- (a) Making use of appropriate object-oriented programming terminology, describe the effect of the following line of code.

3

```
DECLARE newPlayer INITIALLY Player("French", "Player1626")
```

[Turn over

## 4. (continued)

- (b) A player's power rating is calculated by working out the average of the speed, jump, shoot and size points.



Using a programming language with which you are familiar, write code to implement the `calcPower()` method.

2

## 4. (continued)

- (c) Players can join a league with nine other players. The league table is sorted in descending order of power.

Player1626	18.5
Player1762	17.25
Player2262	16.0
Player3562	14.75
Player1092	14.5
Player1827	12.75
Player1468	12.25
Player8372	11.5
Player6391	9.5
Player2864	9.0

- (i) Using appropriate object-oriented terminology, explain the purpose of the code below.

2

```
DECLARE league AS ARRAY OF Player INITIALLY [Null] * 10
SET league[0] TO newPlayer
```

- (ii) The incomplete code below is used to arrange the league table details in descending order of power by applying a bubble sort algorithm to the league variable defined in part (i) above.

```

Line 2001    DECLARE numPlayers INITIALLY 9
Line 2002    DECLARE swapped INITIALLY TRUE
Line 2003    WHILE _____
Line 2004        SET swapped TO FALSE
Line 2005        FOR loop FROM 0 TO numPlayers-1 DO
Line 2006            IF _____
Line 2007                SET temp TO league[loop]
Line 2008                SET league[loop] TO league[loop +1]
Line 2009                SET league[loop+1] TO temp
Line 2010        END IF
Line 2011    END FOR
Line 2012    SET numPlayers TO numPlayers - 1
Line 2013    END WHILE
```

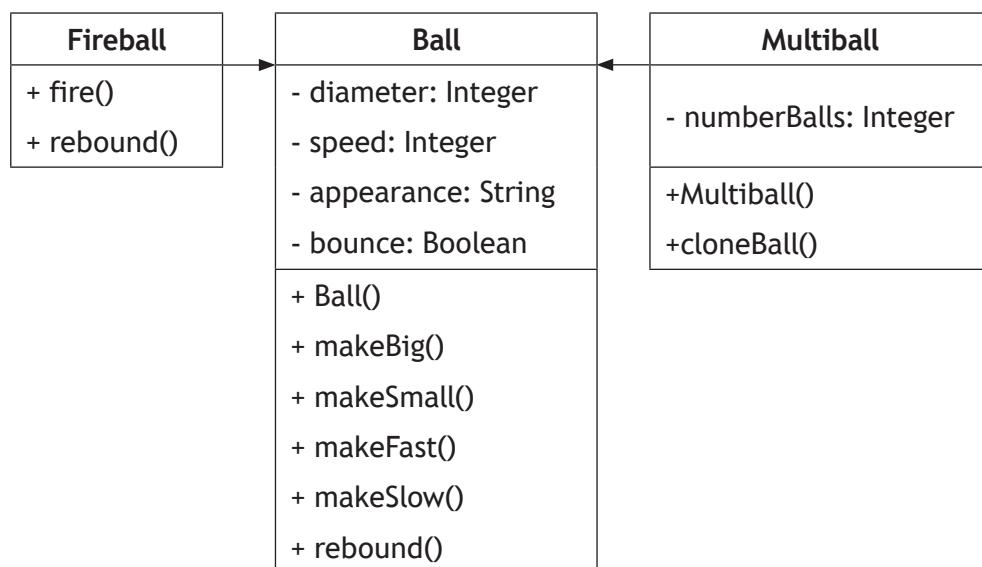
Using a programming language of your choice, write the code needed to complete Lines 2003, 2006 and 2010.

3

**4. (continued)**

- (d) During the game, players can collect power-ups that allow them to change the type of ball being used. For example, the ball can become bigger or smaller, faster or slower, it can become an unstoppable fire ball or split into multiple balls.

The UML class diagram below shows the classes used to represent each ball type that is possible.



- (i) By referring to details in the UML class diagram above, explain what is meant by inheritance.
- (ii) The code used to define the `rebound()` method in the `Ball` and `Fireball` classes is shown below.

1

Ball Method

```

PROCEDURE rebound()
  SET THIS.bounce TO TRUE
END PROCEDURE
  
```

Fireball Method

```

OVERRIDE PROCEDURE rebound()
  SET THIS.bounce TO FALSE
END PROCEDURE
  
```

Name the feature of object-oriented programming illustrated by this example.

1