Practice Test 3

AP® Computer Science A Exam

SECTION I: Multiple-Choice Questions

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO.

At a Glance

Total Time
1 hour 30 minutes
Number of Questions
40
Percent of Total Score
50%
Writing Instrument
Pencil required

Instructions

Section I of this examination contains 40 multiple-choice questions. Fill in only the ovals for numbers 1 through 40 on your answer sheet.

Indicate all of your answers to the multiple-choice questions on the answer sheet. No credit will be given for anything written in this exam booklet, but you may use the booklet for notes or scratch work. After you have decided which of the suggested answers is best, completely fill in the corresponding oval on the answer sheet. Give only one answer to each question. If you change an answer, be sure that the previous mark is erased completely. Here is a sample question and answer.

Sample Question	Sample Answer
Chicago is a (A) state (B) city (C) country (D) continent (E) county	

Use your time effectively, working as quickly as you can without losing accuracy. Do not spend too much time on any one question. Go on to other questions and come back to the ones you have not answered if you have time. It is not expected that everyone will know the answers to all the multiple-choice questions.

About Guessing

Many candidates wonder whether or not to guess the answers to questions about which they are not certain. Multiple-choice scores are based on the number of questions answered correctly. Points are not deducted for incorrect answers, and no points are awarded for unanswered questions. Because points are not deducted for incorrect answers, you are encouraged to answer all multiple-choice questions. On any questions you do not know the answer to, you should eliminate as many choices as you can, and then select the best answer among the remaining choices.

GO ON TO THE NEXT PAGE.

Java Quick Reference

Class Constructors and Methods	Explanation
String Class	
`String(String str)`	Constructs a new `String` object that represents the same sequence of characters as `str`
`int length()`	Returns the number of characters in a `String` object
`String substring(int from, int to)`	Returns the substring beginning at index `from` and ending at index `to - 1`
`String substring(int from)`	Returns `substring(from, length())`
`int indexOf(String str)`	Returns the index of the first occurrence of `str`; returns `-1` if not found
`boolean equals(String other)`	Returns `true` if `this` is equal to `other`; returns `false` otherwise
`int compareTo(String other)`	Returns a value <0 if `this` is less than `other`; returns zero if `this` is equal to `other`; returns a value of >0 if `this` is greater than `other`
Integer Class	
`Integer(int value)`	Constructs a new `Integer` object that represents the specified `int` value
`Integer.MIN_VALUE`	The minimum value represented by an `int` or `Integer`
`Integer.MAX_VALUE`	The maximum value represented by an `int` or `Integer`
`int intValue()`	Returns the value of this `Integer` as an `int`
Double Class	
`Double(double value)`	Constructs a new `Double` object that represents the specified `double` value
`double doubleValue()`	Returns the value of this `Double` as a `double`
Math Class	
`static int abs(int x)`	Returns the absolute value of an `int` value
`static double abs(double x)`	Returns the absolute value of a `double` value
`static double pow(double base, double exponent)`	Returns the value of the first parameter raised to the power of the second parameter
`static double sqrt(double x)`	Returns the positive square root of a `double` value
`static double random()`	Returns a `double` value greater than or equal to `0.0` and less than `1.0`
ArrayList Class	
`int size()`	Returns the number of elements in the list
`boolean add(E obj)`	Appends `obj` to end of list; returns `true`
`void add(int index, E obj)`	Inserts `obj` at position index (`0 <= index <= size`), moving elements at position `index` and higher to the right (adds `1` to their indices) and adds 1 to size
`E get(int index)`	Returns the element at position `index` in the list
`E set(int index, E obj)`	Replaces the element at position `index` with `obj`; returns the element formerly at position `index`
`E remove(int index)`	Removes the element at position `index`, moving elements at position `index + 1` and higher to the left (subtracts `1` from their indices) and subtracts 1 from size; returns the element formerly at position `index`
Object Class	
`boolean equals(Object other)`	
`String toString()`	

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE A

SECTION I

Time—1 hour and 30 minutes

Number of Questions—40

Percent of total exam grade—50%

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in the method calls are not `null` and that methods are called only when their preconditions are satisfied.

1. What will be the result of the following statement?

```
(int)(5 / 2)
```

(A) `2`
(B) `2.5`
(C) `3`
(D) `5`
(E) The statement would generate an error.

2. Which of the following statements is equivalent to `num += 1`?

(A) `num = 1`
(B) `num + 1`
(C) `num++1`
(D) `1 += num`
(E) `num = num + 1`

3. Which of the following would correctly generate a random number between 5 and 25 (inclusive)?

(A) `(int) (Math.random( ) * (20)`
(B) `(int) (Math.random( ) * (21)`
(C) `(int) (Math.random( ) * (21) + 5`
(D) `(int) (Math.random( ) * (25)`
(E) `(int) (Math.random( ) * (25) + 5`

GO ON TO THE NEXT PAGE.

4. What would be stored in the difference variable based on the following code?

```
String city1 = "Chicago";
String city2 = "London";
int difference = city1.compareTo(city2);
```

(A) `0`
(B) `1`
(C) A number less than `0`.
(D) A number greater than `0`.
(E) The code would generate an error.

5. Assuming `city1` and `city2` are variables of type String, initialized appropriately with unknown values, what would be printed based on the following code?

```
if (city1.compareTo(city2) > 0)
   print(city1);
else
   print(city2);
```

(A) The city that comes first alphabetically.
(B) The city that comes second alphabetically.
(C) Both cities.
(D) The city that comes first alphabetically, or a `NullPointerException` if either variable was not initialized.
(E) The city that comes second alphabetically, or a `NullPointerException` if either variable was not initialized.

6. Which of the following expressions is equivalent to `!(x < 5 && y > 10)` ?

(A) `x >= 5 && y <= 10`
(B) `x <= 5 && y >= 10`
(C) `x >= 5 || y <= 10`
(D) `x <= 5 || y >= 10`
(E) `!x < 5 && !y > 10`

7. What is the variable `result` equal to after the execution of the following code?

```
int num = 0;
boolean result = (num == 0) || (10 / num < 5);
```

(A) `0`
(B) `true`
(C) `false`
(D) `10`
(E) The code would generate an error.

GO ON TO THE NEXT PAGE.

8. Which of the following `if` statements will run below?

```
String president = "Lincoln";
String city = "Lincoln";

if (city == president)
  // Statement 1
else if (city.equals(president))
  // Statement 2
else
  // Statement 3
```

(A) Statement 1
(B) Statement 2
(C) Statement 3
(D) Statement 1 and Statement 2
(E) Statement 1, Statement 2, and Statement 3

9. What will `num` equal after the following code?

```
int num = 0;
if (num < 10)
        num++;
else if (num == 1)
        num++;
else
        num = 10;
```

(A) `0`
(B) `1`
(C) `2`
(D) `3`
(E) `10`

10. What will `num` equal after the following code?

```
int num = 0;
if (num < 10)
        num++;
if (num == 1)
        num++;
else
        num = 10;
```

(A) `0`
(B) `1`
(C) `2`
(D) `3`
(E) `10`

GO ON TO THE NEXT PAGE.

11. What will num equal after the following code?

```
int num = 0;
if (num < 10)
        num++;
else if (num == 1)
        num++;
if (num == 2)
        num++;
else
        num = 10
```

(A) 0
(B) 1
(C) 2
(D) 3
(E) 10

12. What will the variable num equal after the following code?

```
int num = 0;
while (num < 10)
        num++;
```

(A) 0
(B) 1
(C) 9
(D) 10
(E) 11

13. What will the variable num equal after the following code?

```
int num = 0;
while (num < 10)
  if (num % 2 == 0)
          num++;
  else
          num *= 2
```

(A) 9
(B) 10
(C) 14
(D) 20
(E) An infinite loop occurs.

GO ON TO THE NEXT PAGE.

14. What condition could replace `*/ Condition Here */` to ensure `num` equals `4` after the code below executes?

```
int num = 1;
while (*/ Condition Here */)
        num *= 2;
```

(A) `num == 4`
(B) `num < 4`
(C) `num <= 4`
(D) `num > 4`
(E) `num >= 4`

15. What condition could replace `*/ Condition Here */` to ensure `num` equals `10` after the code below executes?

```
int num = 0;
for (int i = 0; */ Condition Here */; i++)
          num += 2;
```

(A) `i < 10`
(B) `i <= 10`
(C) `num < 10`
(D) `num <= 10`
(E) `i == num`

16. How many times will the `for` loop body below execute?

```
for (int i = 0; i <= 10; i += 2)
         // for loop body
```

(A) 0
(B) 5
(C) 6
(D) 10
(E) 11

17. How many times will the `for` loop body below execute?

```
for (int i = 1000; i > 0; i--)
         // for loop body
```

(A) 0
(B) 999
(C) 1000
(D) 1001
(E) An infinite loop will occur.

GO ON TO THE NEXT PAGE.

18. What will the variable `num` equal after the following code executes?

```
int num = 0;
for (int i = 0; i < 5; i++)
        for (int k = 0; k < 2; k++)
                num++;
```

(A) 0
(B) 2
(C) 5
(D) 7
(E) 10

19. What will `secondString` be equal to after the following code executes?

```
String firstString = "mississippi";
String secondString = "";
for (int i = 0; i < 5; i++)
  secondString += firstString.substring(i, i + 1);
```

(A) ""
(B) "miss"
(C) "missi"
(D) "msisipi"
(E) "mississippi"

20. What will be printed by the following code?

```
String[ ] cities ={"Chicago", "London", "Tokyo", "Manila"};
for (String city: cities)
       System.out.println(city);
```

(A) Chicago
(B) ChicagoLondonTokyoManila
(C) Chicago
    London
    Tokyo
    Manila
(D) Manila
    Tokyo
    London
    Chicago
(E) Nothing will be printed.

GO ON TO THE NEXT PAGE.

The following class will be used for questions 21-23.

```
public class BaseballPlayer
{
    String firstName;
    String lastName;
    int number;
    String position;

    public BaseballPlayer(String first, String last, int num) {
            // Constructor
    }

    public void setPosition(String pos) {
            // Implementation not shown
    }
}
```

21. The constructor needs to initialize the `firstName`, `lastName`, and `number` variables. What code would be used to complete the constructor?

(A)
```
first = firstName;
last = lastName;
num = number;
```
(B)
```
firstName = first;
lastName = last;
number = num;
```
(C)
```
String first = firstName;
String last = lastName;
int num = number;
```
(D)
```
String firstName = first;
String lastName = last;
int number = num;
```
(E)
```
String first = String firstName;
String last = String lastName;
int num = int number;
```

22. How would you implement the `setPosition` mutator to set the `position` variable of a `BaseballPlayer`?

(A) `return pos;`
(B) `return position;`
(C) `pos = position;`
(D) `position = pos;`
(E) `String pos = position;`

23. If the parameters of the constructor were called `firstName`, `lastName`, and `number`, what could you do to initialize the class-level variables?

(A) Change the names of the parameters.
(B) Change the names of the class-level variables.
(C) Use the `this` keyword in front of the parameters within the constructor.
(D) Use the `this` keyword in front of the class-level variables within the constructor.
(E) It isn't possible to initialize the class-level variables if the parameters have the same name.

GO ON TO THE NEXT PAGE.

24. Which of the following correctly creates a `String` array of length 10?

(A) `String myArray = new String[10];`
(B) `myArray String = new String[10];`
(C) `String myArray = new array[10];`
(D) `String[ ] myArray = new array[10];`
(E) `String[ ] myArray = new String[10];`

25. What does the following code do, assuming `players` is an array of `Strings`?

```
String temp = players[0];
for (int i = 1; i < players.length; i++)
        if (players[i].compareTo(temp) < 0)
                temp = players[i];
```

(A) Find `Strings` in the array that are equal to `temp`.
(B) Find the `String` in the array that comes first alphabetically and store it in `temp`.
(C) Find the `String` in the array that comes last alphabetically and store it in `temp`.
(D) Find the first `String` in the array that comes before `temp` alphabetically and store it in `temp`.
(E) Find the first `String` in the array that comes after `temp` alphabetically and store it in `temp`.

26. What does the following code do, assuming `nums` is an array of integers?

```
int x = 0;
for (int i = 0; i < nums.length; i++)
        if (nums[i] % 2 == 0)
                x++;
```

(A) Count the number of items in `nums`.
(B) Get the total of all the numbers in `nums`.
(C) Get the total of all the even numbers in `nums`.
(D) Count the number of even numbers in `nums`.
(E) Count the number of odd numbers in `nums`.

27. How would you access the 10th element of an array called `myArray`?

(A) `myArray[9]`
(B) `myArray[10]`
(C) `myArray = 9`
(D) `myArray = 10`
(E) `[10]myArray`

28. Which of the following is required to use binary search?

(A) The data must be stored in an array.
(B) The data must be stored in an `ArrayList`.
(C) The data must be sorted.
(D) The data must be integers.
(E) Both A and C are required to use binary search.

GO ON TO THE NEXT PAGE.

29. If an `ArrayList` has the following elements and each step of sorting is shown, what sorting method is being used?

Original List:	42 25 10 20 9
Step 1:	9 25 10 20 42
Step 2:	9 10 25 20 42
Step 3:	9 10 20 25 42
Step 4:	9 10 20 25 42

(A) Bubble Sort
(B) Insertion Sort
(C) Merge Sort
(D) Quick Sort
(E) Selection Sort

30. What will be stored in the variable `x` after the following code executes, assuming `myList` is an `ArrayList` of `Strings`?

```
int x = 0;
for (String temp: myList)
       if (temp.substring(0, 1).equals("s"))
              x++;
```

(A) The number of elements in `myList` equal to `"s"`.
(B) The number of elements in `myList` that start with `"s"`.
(C) The number of elements in `myList` that end with `"s"`.
(D) The total number of elements in `myList`.
(E) `x` will remain equal to `0`.

31. What is the correct way to declare and initialize a 2D array of `Strings` with 5 rows and 10 columns?

(A) `String myArray = new array[10][5];`
(B) `String myArray = new String[5][10];`
(C) `String[ ] myArray = new String[10][5];`
(D) `String[ ][ ] myArray = new String[5][10];`
(E) `String[ ][ ] myArray = new String[10][5];`

32. What will be stored in the variable `x` after the following code executes, assuming `myArray` is a 2D array of integers?

```
int x = 0;
for (int[ ] row: myArray)
         for (int temp: row)
                x += temp;
```

(A) The total number of elements in the 2D array.
(B) The total number of rows in the 2D array.
(C) The total number of columns in the 2D array.
(D) The sum of all of the elements in the 2D array.
(E) The sum of the elements in the first row of the 2D array.

GO ON TO THE NEXT PAGE.

33. What should replace `/* condition */` to correctly go through the entire 2D array called `myArray`?

```
for (int row = 0; row <= myArray.length; row++)
        for (int col = 0; col <= /* condition */; col++)
```

(A) `myArray.length`
(B) `myArray.length - 1`
(C) `myArray[0].length`
(D) `myArray[0].length - 1`
(E) `myArray[length - 1]`

34. What would the class header look like for a class called `Teacher` that is a subclass of `Employee`?

(A) `public class Teacher`
(B) `public class Teacher extends Employee`
(C) `public class Teacher implements Employee`
(D) `public class Employee extends Teacher`
(E) `public class Employee implements Teacher`

35. The constructor below has an error. Which line contains the error?

```
1    public void Teacher(String name, int salary) {
2         super(name);
3         this.salary = salary;
     }
```

(A) Line 1
(B) Line 2
(C) Line 3
(D) Lines 1 and 2
(E) Lines 2 and 3

36. If you create a class that has a method called `calculatePerimeter` that overrides a superclass method of the same name, what object-oriented programming concept is this an example of?

(A) Abstraction
(B) Class Design
(C) Encapsulation
(D) Inheritance
(E) Polymorphism

37. Which of the following is the superclass for all classes in Java?

(A) `Class`
(B) `Comparable`
(C) `Java`
(D) `Object`
(E) `String`

GO ON TO THE NEXT PAGE.

38. Which line contains the base case in the recursive code below?

```
1    public int sum(int num) {
2          int total;
3          if (num == 1)
4                  total = 1;
5          else
6                  total = num + sum(num - 1);
7          return total;
     }
```

(A) Line 2
(B) Line 3
(C) Line 4
(D) Line 6
(E) Line 7

39. What would be the result of the call `sum(5)`?

```
public int sum(int num) {
        int total;
        if (num == 1)
                total = 1;
        else
                total = num + sum(num - 1);
        return total;
}
```

(A) 4
(B) 5
(C) 10
(D) 14
(E) 15

40. How many recursive calls would occur with the call `fib(5)`?

```
public int fib(int num) {
       if (num <= 1)
               return n;
       else
               return fib(n - 1) + fib(n - 2);
}
```

(A) 0
(B) 1
(C) 5
(D) 10
(E) 15

END OF SECTION I

IF YOU FINISH BEFORE TIME IS CALLED,
YOU MAY CHECK YOUR WORK ON THIS SECTION.

DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.

COMPUTER SCIENCE A

SECTION II

Time—1 hour and 30 minutes

Number of Questions—4

Percent of Total Grade—50%

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA™.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

FREE-RESPONSE QUESTIONS

1. A day care has a program to keep track of its employees and which children they teach during the day. An `Employee` has a minimum and maximum age they can teach. The `DayCare` also has a maximum ratio that specifies the maximum number of children a single employee can teach. Below is the full `DayCare` class:

```
public class DayCare
{
    private ArrayList<Employee> employees;
    private ArrayList<Child> children;
    private int maxRatio;

    public DayCare(int maxRatio)
    {
       employees = new ArrayList<Employee>();
       children = new ArrayList<Child>();
       this.maxRatio = maxRatio;
    }

    public boolean findEmployeeForChild(Child c)
    {
       /* To be completed in part (a) */
    }

    public boolean runDayCare()
    {
       /* To be completed in part (b) */
    }

    public boolean addChild(Child c)
    {
       /* To be completed in part (c) */
    }
}
```

GO ON TO THE NEXT PAGE.

(a) An `Employee` can only teach children between the employee's minimum age (inclusive) and maximum age (inclusive). They can also only teach children up to the day care's maximum ratio (inclusive). Below is the full `Employee` class.

```
public class Employee
{
    /* Instance variables not shown */

    public Employee(String name, String id, int min, int max)
    {
        /* Implementation not shown */
    }

    // Return the number of children currently assigned to this Employee
    public int childrenAssigned()
    {
        /* Implementation not shown */
    }

    // Assign a new child to this Employee
    public void assignChild(Child c)
    {
        /* Implementation not shown */
    }

    // Determine whether this Employee can teach a Child based on the child's age
    public boolean canTeach(int age)
    {
        /* Implementation not shown */
    }
}
```

A `Child` has accessors to get their name and age. While the implementation of `Child` is not shown, you can assume the accessors are called `getName` and `getAge`.

Complete the `findEmployeeForChild` method below that assigns a `Child` to the first `Employee` who can teach the `Child` and who has not reached the maximum ratio of the `DayCare`.

```
/* Return true if an Employee was found for the Child, false otherwise */
public boolean findEmployeeForChild(Child c)
```

GO ON TO THE NEXT PAGE.

(b) In order for the `DayCare` to run for a day, each `Child` must be assigned an `Employee`. If an `Employee` cannot be found for a `Child`, the `DayCare` cannot run for the day.

Complete the `runDayCare` method below that finds an `Employee` for each `Child` in the `children ArrayList`.

```
/* Return true if an Employee was found for each Child, false otherwise */
public boolean runDayCare(Child c)
```

GO ON TO THE NEXT PAGE.

(c) When a `Child` is added to the roster of the `DayCare`, the `DayCare` should first make sure there is an `Employee` available to teach that `Child`.

Complete the `addChild` method below that adds a `Child` to the `children ArrayList` if an `Employee` is available to teach that `Child`.

```
/* Return true if the Child was added to the ArrayList, false otherwise */
public boolean addChild(Child c)
```

GO ON TO THE NEXT PAGE.

2. A baseball team consists of different people including players, coaches, and people who work in the front office making trades and other transactions. The following `Person` class is used for all of the people who work for the team.

Each person has a name and an age.

```
public class Person
{
    private String fullName;
    private int age;

    public Person(String s, int a)
    {
       fullName = s;
       age = a;
    }

    // Accessors for name and age
    public String getName()
    {
       return fullName;
    }

    public int getAge()
    {
       return age;
    }
}
```

GO ON TO THE NEXT PAGE.

A `Player` has a name and age just like any person on the team, but they also have a `position`. The position could be something like "catcher," "left fielder," or "infielder." A `Player` should also be able to change their position using a method called `changePosition`. Here is an example of a `Player` object:

```
Player p = new Player("Sammy Sosa", 32, "right fielder");
p.changePosition("outfielder");
```

Write the entire `Player` class below.

GO ON TO THE NEXT PAGE.

3. A class is designed to store someone's full name. You can assume the name has only one space between the first name and the last name. The class has methods to extract the first name, last name, and number of vowels in the name. You can see an example below.

```
String fullName = "Katherine Johnson";
Name.getFirstName(fullName); // Returns "Katherine"
Name.getLastName(fullName); // Returns "Johnson"
Name.countVowels(fullName); // Returns 6
```

(a) The `getFirstName` method returns the first name based on a given full name. You can assume that full name has only one space between the first name and the last name. Write the `getFirstName` method below.

```
public static String getFirstName(String name)
{




}
```

GO ON TO THE NEXT PAGE.

(b) The `getLastName` method returns the last name based on a given full name. You can assume that full name has only one space between the first name and the last name. Write the `getLastName` method below.

```
public static String getLastName(String name)
{




}
```

GO ON TO THE NEXT PAGE.

(c) The `countVowels` method counts the number of vowels in the given name. You can assume we will only count the letters a, e, i, o, and u as vowels. Write the entire `countVowels` method below.

```
public static int countVowels(String name)
{
```

GO ON TO THE NEXT PAGE.

4. A city parking lot has a sign that keeps track of how many parking spaces are available in the lot. The class for the parking lot is detailed below.

```
public class ParkingLot
{
    private Car[ ][ ] lot;

    public ParkingLot(int rows, int cols)
    {
       lot = new Car[rows][cols];
    }

    public int openSpaces()
    {
       // Complete in part (a)
    }

    public boolean parkCar(Car newCar)
    {
       // Complete in part (b)
    }
}
```

GO ON TO THE NEXT PAGE.

(a) Write the `openSpaces` method that returns how many spaces are available in the parking lot. If a space is empty, it will be equal to `null`.

```
/* Return the number of empty spaces in the parking lot */
public int openSpaces( )
```

GO ON TO THE NEXT PAGE.

(b) Complete the `parkCar` method that puts a new car in any space in the parking lot and returns `true` if was able to do so. It should return `false` if there are no empty spaces. You should use the `openSpaces` method to receive full credit.

```
/* Return true if there is an open spot to park the newCar, false otherwise. The car should be added to the lot 2D array
if there is an open spot. */
public boolean parkCar(Car newCar)
```

STOP

END OF EXAM

The Princeton Review®

Completely darken bubbles with a No. 2 pencil. If you make a mistake, be sure to erase mark completely. Erase all stray marks.

1.

YOUR NAME: _____
(Print) Last First M.I.

SIGNATURE: _____ DATE: ___/___/___

HOME ADDRESS: _____
(Print) Number and Street

_____
City State Zip Code

PHONE NO.: _____

IMPORTANT: Please fill in these boxes exactly as shown on the back cover of your test book.

2. TEST FORM

3. TEST CODE

0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H
8	I
9	

0	A	J	0	0
1	B	K	1	1
2	C	L	2	2
3	D	M	3	3
4	E	N	4	4
5	F	O	5	5
6	G	P	6	6
7	H	Q	7	7
8	I	R	8	8
9			9	9

4. REGISTRATION NUMBER

0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9

5. YOUR NAME

First 4 letters of last name					FIRST INIT	MID INIT
A	A	A	A		A	A
B	B	B	B		B	B
C	C	C	C		C	C
D	D	D	D		D	D
E	E	E	E		E	E
F	F	F	F		F	F
G	G	G	G		G	G
H	H	H	H		H	H
I	I	I	I		I	I
J	J	J	J		J	J
K	K	K	K		K	K
L	L	L	L		L	L
M	M	M	M		M	M
N	N	N	N		N	N
O	O	O	O		O	O
P	P	P	P		P	P
Q	Q	Q	Q		Q	Q
R	R	R	R		R	R
S	S	S	S		S	S
T	T	T	T		T	T
U	U	U	U		U	U
V	V	V	V		V	V
W	W	W	W		W	W
X	X	X	X		X	X
Y	Y	Y	Y		Y	Y
Z	Z	Z	Z		Z	Z

6. DATE OF BIRTH

Month	Day		Year	
JAN				
FEB	0	0	0	0
MAR	1	1	1	1
APR	2	2	2	2
MAY	3	3	3	3
JUN		4	4	4
JUL		5	5	5
AUG		6	6	6
SEP		7	7	7
OCT		8	8	8
NOV		9	9	9
DEC				

7. GENDER

- MALE
- FEMALE

The Princeton Review®

1. A	B	C	D	E
2. A	B	C	D	E
3. A	B	C	D	E
4. A	B	C	D	E
5. A	B	C	D	E
6. A	B	C	D	E
7. A	B	C	D	E
8. A	B	C	D	E
9. A	B	C	D	E
10. A	B	C	D	E
11. A	B	C	D	E
12. A	B	C	D	E
13. A	B	C	D	E
14. A	B	C	D	E
15. A	B	C	D	E
16. A	B	C	D	E
17. A	B	C	D	E
18. A	B	C	D	E
19. A	B	C	D	E
20. A	B	C	D	E
21. A	B	C	D	E
22. A	B	C	D	E
23. A	B	C	D	E
24. A	B	C	D	E
25. A	B	C	D	E
26. A	B	C	D	E
27. A	B	C	D	E
28. A	B	C	D	E
29. A	B	C	D	E
30. A	B	C	D	E
31. A	B	C	D	E
32. A	B	C	D	E
33. A	B	C	D	E
34. A	B	C	D	E
35. A	B	C	D	E
36. A	B	C	D	E
37. A	B	C	D	E
38. A	B	C	D	E
39. A	B	C	D	E
40. A	B	C	D	E