ITPEC
Information Technology Professionals Examination Council

Spring 2011

Fundamental IT Engineer Examination (Afternoon)

Questions must be answered in accordance with the following:

Question Nos.	Q1 – Q6	Q7 , Q8
Question Selection	Compulsory	Select 1 of 2
Examination Time	13:30 – 16:00 (150 minutes)	

Instructions:

1. Use a pencil. If you need to change an answer, erase your previous answer completely and neatly. Wipe away any eraser debris.

2. Mark your examinee information and test answers in accordance with the instructions below. Your answer will not be graded if you do not mark properly. Do not mark or write on the answer sheet outside of the prescribed places.

   (1) **Examinee Number**
   Write your examinee number in the space provided, and mark the appropriate space below each digit.

   (2) **Date of Birth**
   Write your date of birth (in numbers) exactly as it is printed on your examination admission card, and mark the appropriate space below each digit.

   (3) **Question Selection**
   For **Q7** and **Q8**, mark the Ⓢ of the question you select to answer in the “Selection Column” on your answer sheet.

   (4) **Answers**
   Mark your answers as shown in the following sample question.

[Sample Question]

In which month is the spring Fundamental IT Engineer Examination conducted in 2011?

Answer group
a) February b) March c) April d) May

Since the correct answer is “b) March”, mark your answer sheet as follows:

[Sample Answer]

Sample	Ⓐ	●	Ⓒ	Ⓓ

Do not open the exam booklet until instructed to do so.
Inquiries about the exam questions will not be answered.

Notations used for pseudo-language

In questions that use pseudo-language, the following notations are used unless otherwise stated.

[Declaration, comment, and process]

Notation		Description
○		Declares names, types, etc. of procedures, variables, etc.
/* text */		Describes comments in text.
Process	• variable ← expression	Assigns the value of an expression to a variable.
	• procedure(argument, ...)	Calls the procedure and passes/receives argument.
	▲ conditional expression process ▼	Indicates a one-way selection process. If the conditional expression is true, then the process is executed.
	▲ conditional expression process 1 ─ process 2 ▼	Indicates a two-way selection process. If the conditional expression is true, then process 1 is executed. If it is false, then process 2 is executed.
	■ conditional expression process ■	Indicates a pre-test iteration process. While the conditional expression is true, the process is executed repeatedly.
	■ process ■ conditional expression	Indicates a post-test iteration process. The process is executed, and then while the conditional expression is true, the process is executed repeatedly.
	■ variable: init, cond, incr process ■	Indicates an iteration process. The initial value init (given by an expression) is stored in the variable at the start of the processing, and then while the conditional expression cond is true, the process is executed repeatedly. An increment incr (given by an expression) is added to the variable in each iteration.

[Logical constants]

`true, false`

(continued on next page)

[Operators and their priorities]

Type of operation	Operator	Priority
Unary operation	+, −, not	High
Multiplication, division	×, ÷, %	
Addition, subtraction	+, −	
Relational operation	>, <, ≥, ≤, =, ≠	
Logical product	and	
Logical sum	or	Low

Note: With division of integers, integer quotient is returned as a result.
The % operator indicates a remainder operation.

Questions **Q1** through **Q6** are all **compulsory**. Answer every question.

Q1. Read the following description concerning hard disk drives, and then answer Subquestions 1 and 2.

Company *X* produces hard disk drives, Products *A* and *B*. The following table shows the specifications for Products *A* and *B*.

Table Specifications for Products *A* and *B*

Specifications	Product *A*	Product *B*
Storage capacity	60 G bytes	80 G bytes
Rotational speed	4,200 rpm	5,400 rpm
Average seek time	14.0 msec	12.0 msec
Data transfer rate	20 M bytes/sec	30 M bytes/sec

Note: 1 G bytes = 1,000 M bytes, 1 M bytes = 1,000 k bytes

Subquestion 1

From the answer group below, select the correct answer to be inserted into each blank ☐ in the following description. The answers shall be rounded off to the first decimal place.

The average rotational latency for Product *A* is $\boxed{\text{A}}$ milliseconds, and it takes $\boxed{\text{B}}$ milliseconds to transfer 100 k bytes of data.
On the other hand, the average rotational latency for Product *B* is 5.5 milliseconds, and it takes 3.3 milliseconds to transfer 100 k bytes of data. Therefore, the average access time (average seek time + average rotational latency + data transfer time) for 100 k bytes of data is $\boxed{\text{C}}$ milliseconds shorter than that for Product *A*.

Answer group

a) 3.3	b) 5.0	c) 5.3	d) 5.5
e) 7.1	f) 11.1	g) 14.2	

Subquestion 2

From the answer group below, select the correct answer to be inserted into the blank $\boxed{\quad\quad}$ in the following description. The answer shall be rounded off to the first decimal place.

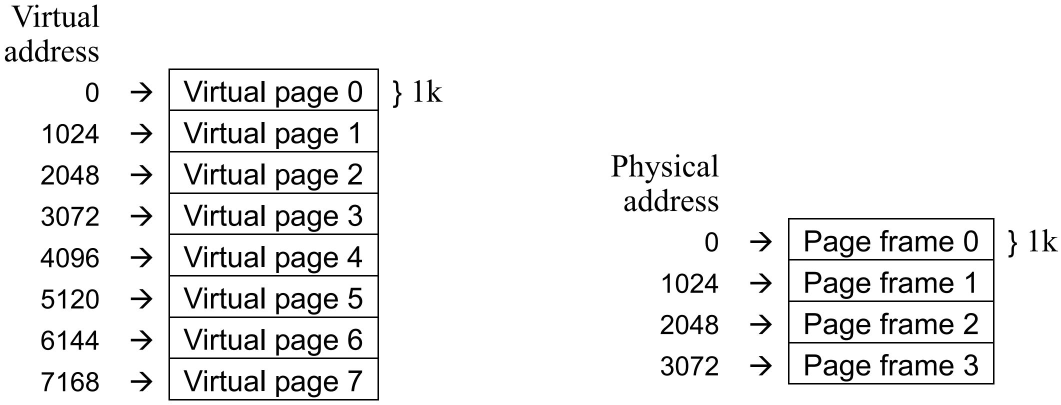
In order to reduce the access time at reading, Product *C*, which is equipped with a disk cache, was developed. It writes to and reads from the disk cache in a block unit of 2 k bytes. When it receives a request to read a data block, it searches the disk cache. If it finds the data block in the disk cache, it returns the data from the disk cache. If it cannot find the data block in the disk cache, it reads data from the magnetic disk and returns the data. At the same time, it replaces the oldest data block in the disk cache with that data block.
The average time to search the disk cache is 1.0 msec. The average hit ratio in the disk cache is 0.4. The average time to read a data block from the disk cache is 0.4 msec. When there is no data cache hit, the average time to read a data block from the magnetic disk is 17.0 msec, including the time to replace the oldest data block in the disk cache. In this case, the average access time to read a block by Product *C* is $\boxed{\quad\text{D}\quad}$ msec.

Answer group

a) 7.0
b) 8.0
c) 10.3
d) 11.3
e) 13.2
f) 14.2

Q2. Read the following description concerning virtual memory controls, and then answer Subquestion.

There is a computer with address space of 8k bytes (virtual addresses: 0 to 8191) and a physical memory (main memory) of 4k bytes (physical addresses: 0 to 4095). Its virtual memory has virtual page size of 1024 bytes, 8 virtual pages, and its physical memory has page frame size of 1024 bytes, 4 page frames. Figure 1(a) illustrates the way to divide up the address space of 8k bytes into 8 virtual pages, and Figure 1(b) illustrates the way to divide up the physical memory of 4k bytes into 4 page frames.



a) Address space of 8k bytes is divided into 8 virtual pages

b) Physical memory of 4k bytes is divided into 4 page frames

Figure 1. Virtual address and physical address

This computer has to organize and maintain a page table so that when a program tries to reference its memory, it has only to generate a 13-bit address corresponding to a virtual address ranges from 0 to 8191, in spite of the fact that physical address only ranges from 0 to 4095.

The 13-bit address has the format as illustrated in Figure 2. In this figure, the 13-bit address is 3077_{10} (= 0 1100 0000 0101_2 = $C05_{16}$), which is regarded as address offset 5 of virtual page 3.

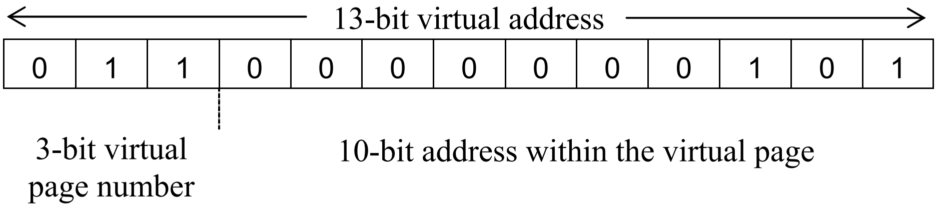

13-bit virtual address
0 1 1 0 0 0 0 0 0 0 1 0 1
3-bit virtual page number
10-bit address within the virtual page

Figure 2. Format of 13-bit address

The page table has the number of entries as much as the number of virtual pages, and each entry has the format as illustrated in Figure 3. It contains the following 3 fields:

1) Present/Absent (leftmost 1 bit): If the page is in physical memory, the bit is set to 1. Else, the bit is set to 0.
2) Secondary storage address (subsequent 12 bits): When the Present/Absent bit is 0, this field has the secondary storage address where the concerned virtual page is saved.
3) Page frame (rightmost 3 bits): When the Present/Absent bit is 1, this field has the page frame number (0 to 3) where the virtual page is loaded.

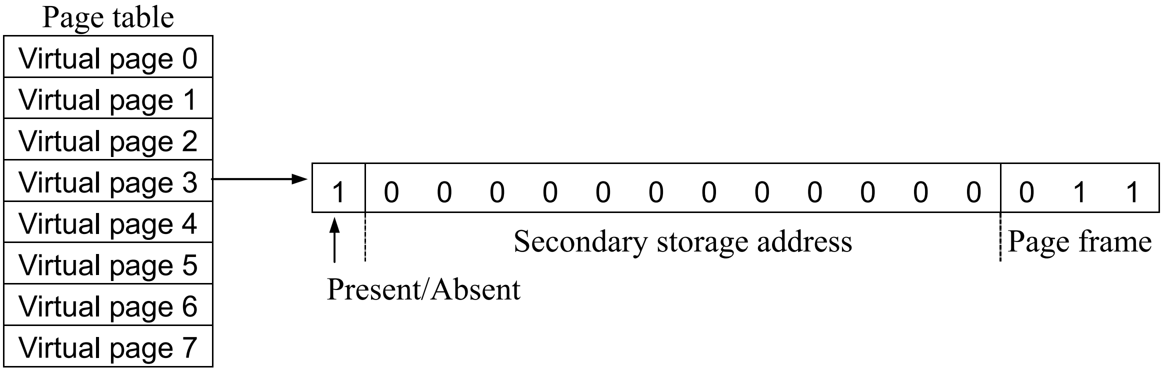
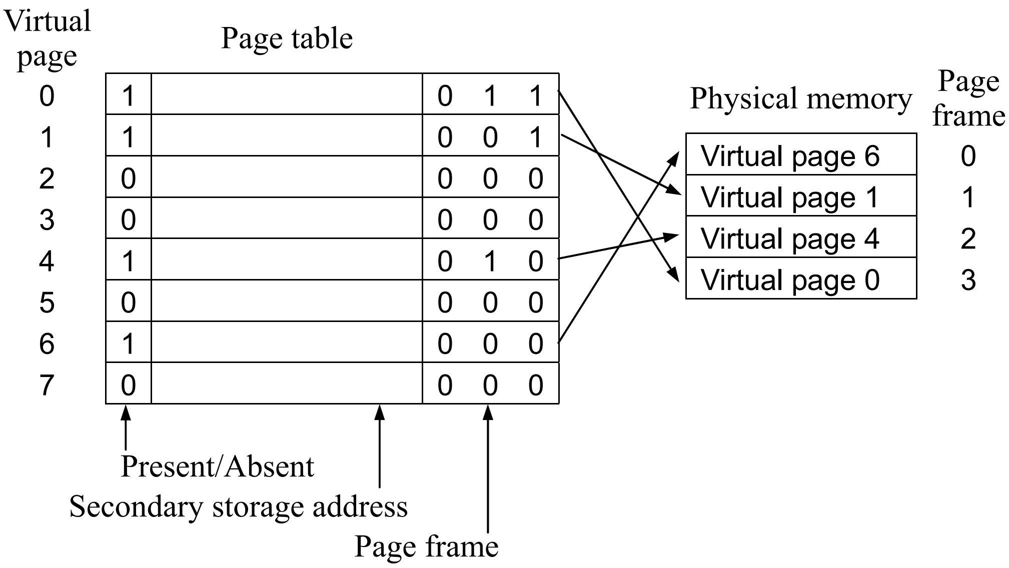


Figure 3. Format of a page table entry

Subquestion

Figure 4 shows the contents of the page table at a certain time of operation. In Figure 4, the contents of the first and third fields are shown in binary numbers, but the contents of the second field are omitted.



Virtual page	Present/Absent	Secondary storage address	Page frame
0	1		0 1 1
1	1		0 0 1
2	0		0 0 0
3	0		0 0 0
4	1		0 1 0
5	0		0 0 0
6	1		0 0 0
7	0		0 0 0

Physical memory	Page frame
Virtual page 6	0
Virtual page 1	1
Virtual page 4	2
Virtual page 0	3

Figure 4. Contents of the page table at a certain time of operation

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the following description.

(1) When the contents of the page table are as shown in Figure 4, a program is trying to update virtual addresses 1020_{10} to 1029_{10}, 2040_{10} to 2049_{10}, and 4090_{10} to 4099_{10} (total 30 bytes). Update is done 1 byte at a time, from lower address to higher address. During program execution, the first $\boxed{\text{A}}$ occurs at virtual address $\boxed{\text{B}}_{10}$. Assuming that no other programs are running in parallel.

(2) The system uses LRU(Least Recently Used) algorithm to determine a virtual page that should be moved out to secondary memory. When the contents of the page table are as shown in Figure 4, a program is trying to update virtual pages 0, 1, 4, 6, 0, 5, 3, 2, 3, 4, 2, 6, in this sequence. As a result of program execution, page frame 0 contains virtual page $\boxed{\text{C}}$, page frame 1 contains virtual page $\boxed{\text{D}}$, and the total number of virtual page replacement in physical memory is $\boxed{\text{E}}$. Assuming that no other programs are running in parallel.

Answer group for A

a) I/O error
b) I/O interruption
c) page fault
d) page fragmentation

Answer group for B

a) 1020
b) 1024
c) 2040
d) 2048
e) 4090
f) 4096

Answer group for C through E

a) 0
b) 1
c) 2
d) 3
e) 4
f) 5
g) 6
h) 7

Q3. Read the following description concerning a database, and then answer Subquestions 1 through 3.

An online shop for DVDs (video films) uses the following three tables:

(1) DVD table

DVD table contains the information about each DVD which the online shop handles. Each DVD has a unique number DID, Title, Category and Price. The table does not contain any null values.

DVD Table

DID	TITLE	CATEGORY	PRICE
1	Snow White and the Seven Dwarfs	Animated Cartoons	15.99
2	Gone with the Wind	Romance	45.90
3	Star Wars Trilogy	Science Fiction	19.50
4	Back to the Future	Science Fiction	12.45
5	Titanic	Romance	39.99
6	Toy Story	Animated Cartoons	15.99

(2) AUDIOTRACK Table

DVDs which the online shop handles often contain several audio tracks in different languages. AUDIOTRACK table contains this information. DID is a foreign key that refers to DVD table. DVDs in the table have one or more audio tracks in different languages. For example, "Snow White and the Seven Dwarfs" has English and Japanese audio tracks.

AUDIOTRACK Table

DID	LANGUAGE
1	English
1	Japanese
2	English
3	English
3	Mandarin
4	English
4	Japanese
4	Hindi
5	English
5	French
6	English
6	Spanish

(3) RATING Table

RATING table contains the ratings of the DVDs rated by customers of the Website. A customer can rate a DVD by giving between one and five stars. DID is a foreign key that refers to DVD table.

RATING Table

DID	UID	STARS	TEXT
1	Bob	3	nice
1	Nina	4	my daughter loves it
2	Lisa	5	I have watched the film 10 times
3	David	4	enjoyed the ending
4	Nicole	4	exciting
4	Craig	3	really good
5	Lisa	2	ok, but picture quality is not good
6	Danny	4	funny

Subquestion 1

The following SQL statement outputs the title, category and average evaluation (number of stars) for all DVDs that have been evaluated by at least 2 customers. The output column name for the average evaluation should have the name GRADE.

From the answer groups below, select the correct answer to be inserted into each blank □ in the following SQL statement.

```
SELECT [     A     ]
FROM DVD D, RATING R
WHERE D.DID = R.DID
[     B     ] BY D.DID, TITLE, CATEGORY
[     C     ] COUNT(*) >= 2
```

Answer group for A

a) TITLE, CATEGORY

b) TITLE, CATEGORY, AVG(STARS)

c) TITLE, CATEGORY, AVG(STARS) AS GRADE

d) TITLE, CATEGORY, GRADE

Answer group for B and C

a) GROUP	b) HAVING
c) ORDER	d) SORT

Subquestion 2

The following SQL statement outputs the title, price and the type of language, “X” for Japanese and “ ” for other languages, for the DVDs.

From the answer group below, select the correct output produced by the following SQL statement.

```
SELECT TITLE, PRICE, ’X’ JAPANESE
FROM DVD D, AUDIOTRACK A
WHERE D.DID = A.DID
AND LANGUAGE = ‘Japanese’
UNION ALL
SELECT TITLE, PRICE, ’ ’ JAPANESE
FROM DVD D
WHERE NOT EXISTS (SELECT *
   FROM AUDIOTRACK A
   WHERE D.DID = A.DID
   AND LANGUAGE = ‘Japanese’)
ORDER BY PRICE DESC
```

Answer group

a)

TITLE	PRICE	JAPANESE
Snow White and the Seven Dwarfs	15.99	X
Back to the Future	12.45	X

b)

TITLE	PRICE	JAPANESE
Back to the Future	12.45	X
Gone with the Wind	45.90	
Snow White and the Seven Dwarfs	15.99	X
Star Wars Trilogy	19.50	
Titanic	39.99	
Toy Story	15.99	

c)

TITLE	PRICE	JAPANESE
Gone with the Wind	45.90	
Titanic	39.99	
Star Wars Trilogy	19.50	
Snow White and the Seven Dwarfs	15.99	X
Toy Story	15.99	
Back to the Future	12.45	X

d)

TITLE	PRICE	JAPANESE
Snow White and the Seven Dwarfs	15.99	X
Gone with the Wind	45.90	
Star Wars Trilogy	19.50	
Back to the Future	12.45	X
Titanic	39.99	
Toy Story	15.99	

Subquestion 3

From the answer group below, select the correct statement concerning the output produced by the following SQL statement.

```
SELECT TITLE
FROM DVD D
WHERE DID NOT IN (SELECT DID
   FROM AUDIOTRACK
   WHERE LANGUAGE <> ‘English’)
```

Answer group for E and F

a) List of titles that do not have English track.
b) List of titles that have English track and one or more non-English track.
c) List of titles that have English track and only one non-English tracks.
d) List of titles that have only English track.

Q4. Read the following description concerning packet filtering, and then answer Subquestions 1 and 2.

Company *X* has built a network shown in the figure, and uses it to publish its Web site to the Internet and to exchange e-mails.

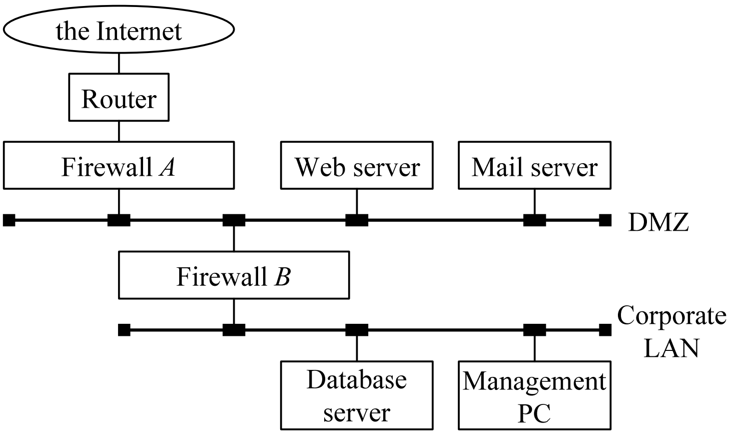


Figure. Network configuration at Company *X*

Company *X*'s network is divided by 2 firewalls into 2 segments: DMZ and corporate LAN. Web server, Mail server, and Database server (hereinafter referred to as DB server) play the following roles:

(1) Web server
As a Web site, Web server publishes information about the company to the Internet. On Web server, a program for processing transactions with external entities is running. Data used by this program is stored on DB server.

(2) Mail server
Mail server sends and receives e-mails to and from external entities. In addition, it runs a program that automatically distributes e-mails to business partners. The data for distributing e-mails is stored on DB server.

(3) DB server
DB server maintains the data used by Web server and Mail server.

From the management PC connected to the corporate LAN, it is possible to login to each server using SSH, and send and receive e-mails to and from external entities via Mail server. Although it is possible to access the corporate Web server from the management PC, access to external Web sites is not permitted.
Table 1 shows the protocols and port numbers used in the network.

Table 1 Protocols and port numbers

Service	Protocol	Port number
Web	HTTP	80
Mail transfer	SMTP	25
Secure shell (remote login)	SSH	22
Mail reception	POP3	110
DB access	Specific to DB	1999

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank □ in the following description. If needed, select the same answer more than once.

Table 2 shows the packet filtering settings for firewall *A*, which connects the Internet and the DMZ. Table 3 shows the packet filtering settings for firewall *B*, which connects the DMZ and the corporate LAN.

Each filtering rule allows or denies communication by specifying the source IP address, destination IP address, and destination port number. Each rule is examined beginning from the top, and the action of the first row whose condition is met is performed. Here, no settings are required for response packets since they are automatically allowed by the dynamic filtering function.

Table 2 Filtering settings for firewall *A*

Condition			Action
Source	Destination	Port number	
Any	Web server	80	Allow
Any	Mail server	25	Allow
A	Any	B	Allow
Any	Any	Any	Deny

Table 3 Filtering settings for firewall *B*

Condition			Action
Source	Destination	Port number	
Web server	DB server	1999	Allow
Mail server	DB server	1999	Allow
Management PC	C	D	Allow
Management PC	Mail server	22	Allow
Management PC	Mail server	25	Allow
Management PC	Web server	80	Allow
Management PC	Web server	22	Allow
Any	Any	Any	Deny

Answer group for A and C

a) Any
b) DB server
c) Mail server
d) Management PC
e) Web server

Answer group for B and D

a) 22
b) 25
c) 80
d) 110
e) 1999

Subquestion 2

In Company *X*'s network, security risks associated with Internet connections are reduced through packet filtering by firewalls, but packet filtering cannot protect the network from all threats. From the answer group below, select the correct answer for the security risks that can be prevented with packet filtering.

Answer group

a) Attacks by unauthorized access from the Internet to the corporate LAN
b) Leakage of files from the company via e-mail
c) SQL injection attacks on the Web site
d) Wiretapping and falsification of data exchanged with the Web site

Q5. Read the following description concerning program design, and then answer Subquestions 1 through 3.

Figure 1 shows the use-case diagram for the Bebe online application. Bebe is a social networking website, where users can post blogs, photographs, music, and videos. Table 1 shows the description of each of the functions shown in Figure 1.

Table 1. Description of functions shown in Figure 1

No.	Description
1.	***Create New Account*** Users first create a new account by specifying some details about themselves, such as user name, password, e-mail address, security question and security question answer.
2.	***Activate Account*** After the new account has been created, a unique account Activation Key will be sent to the users via e-mail. Users then activate their account using the key. Once the new account has been successfully activated, users will be automatically logged in to the system so that they can use the user account management functionality.
3.	***Re-Request Activation Key*** When users do not receive the e-mail with the Activation Key for some reason, the system will provide users with the ability to directly re-request their account Activation Key.
4.	***Login*** When users have an active account (it has been successfully activated), they are free to log in and use the relevant functionality whenever they choose.
5.	***Logout*** This function enables logged in users to directly log off the system when they're finished using it.
6.	***Reset Password*** User's password can be reset to a default password if desired. This function uses the "Change Password" function when overwriting the password with the default one.
7.	***Change Password*** This function enables authenticated users to change their existing passwords. To be protected against unauthorized password changes, users will need to provide their security question answer before they are allowed to change their existing password.
8.	***Update Account Details*** This function provides logged in users with the ability to change their existing account details, including the user name, e-mail address, security question and security question answer. The function does not allow users to change their password, as this is achieved via the "Change Password" function.
9.	***Close Account*** This function allows logged in users to close their account completely. It will allow users to specify comments for why they are closing their account and to request e-mail confirmation that the account has been closed.

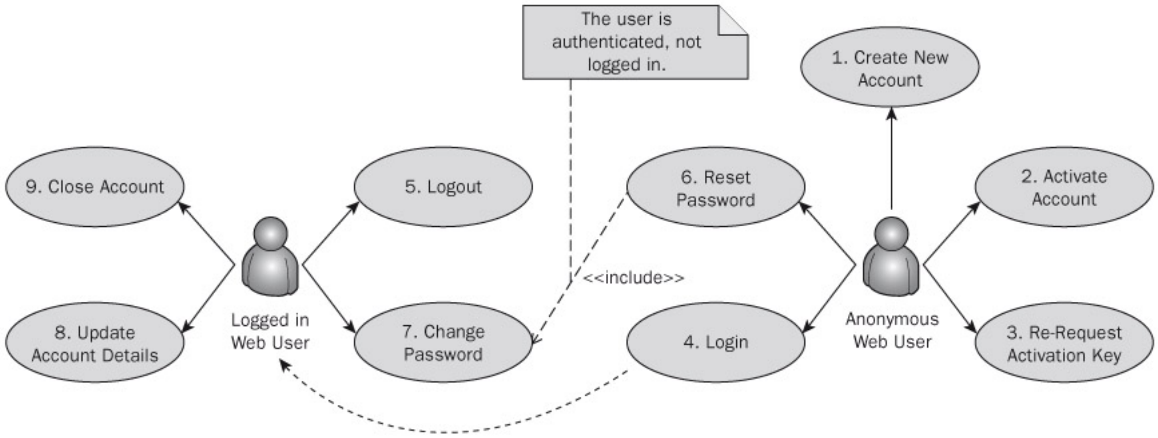


Figure 1. Use-case diagram for Bebe Online Application.

The preceding features and functions will be organized across a series of different Web pages and flows on the site. Each Web page will be responsible for gathering user input and passing it on for back-end processing. Figure 2 shows the high-level navigation map for the site and the associated user stories.

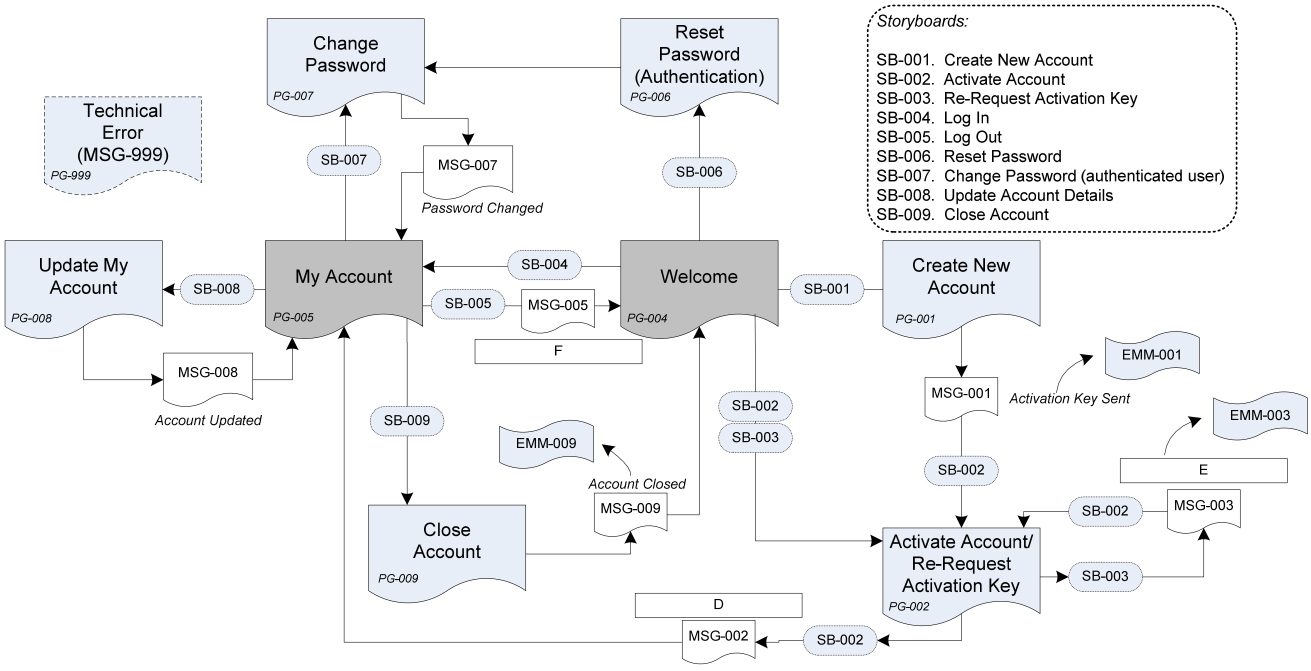


Figure 2. High-level navigation map and associated user stories.

As shown in Figure 2, all the user stories start from either Welcome page or My Account page. The user stories can also flow from one into another – for example, Create New Account flows into Activate Account, which can also flow into Re-Request Activation Key. The navigation map doesn't show all the possible scenarios; it just shows the "normal" flows, the “confirmation” messages that are displayed to the user, and the e-mail messages that are sent to them.

Tables 2 through 5 below show the elements in the navigation map.

Table 2. Story Boards

Story Board No.	Description
SB-001	Create New Account
SB-002	Activate Account
SB-003	Re-Request Activation Key
SB-004	Login
SB-005	Logout
SB-006	Reset Password
SB-007	Change Password
SB-008	Update Account Details
SB-009	Close Account

Table 3. Logical Pages

Page No.	Description
PG-001	Create New Account
PG-002	Activate Account / Re-Request Activation Key
PG-004	Welcome (and Login)
PG-005	My Account (and Logout)
PG-006	Reset Password
PG-007	Change Password
PG-008	Update My Account
PG-009	Close Account
PG-999	Technical Error

Table 4. Display messages

Message No.	Description
MSG-001	Activation Key Sent
MSG-002	D
MSG-003	E
MSG-005	F
MSG-007	Password Changed
MSG-008	Account Updated
MSG-009	Account Closed
MSG-999	Technical Error

Table 5. E-mail messages

Message No.	Description
EMM-001	Account Activation Key
EMM-003	Account Activation Key Re-Request
EMM-009	Account Closed

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank $\boxed{}$ in the following description.

In the case of a “forgotten” password, users can first display Welcome page, but cannot proceed to My Account page directly, because they cannot login. In this case, the routes from Welcome page to My Account page will be:

Welcome → $\boxed{\text{A}}$ → $\boxed{\text{B}}$ → My Account

During this process, users are requested to enter user name, e-mail address and security question answer. For instance, on the page $\boxed{\text{A}}$, users are requested to enter $\boxed{\text{C}}$.

Answer group for A and B

a) Activate Account	b) Change Password
c) Create New Account	d) Re-Request Activation Key
e) Reset Password	f) Update My Account

Answer group for C

a) e-mail address
b) e-mail address and security question answer
c) e-mail address and user name
d) security question answer
e) security question answer and user name
f) user name

Subquestion 2

The function “Activate Account” has one or several routes to it. From the answer group below, select the list of routes that lead to the function “Activate Account”.

Answer group

a) Create New Account
b) Create New Account, Re-Request Activation Key

c) Create New Account, My Account, Re-Request Activation Key

d) Create New Account, Re-Request Activation Key, Welcome

e) Change Password, My Account, Re-Request Activation Key

f) My Account, Re-Request Activation Key, Welcome

Subquestion 3

From the answer group below, select the correct answer to be inserted into each blank D through F in Figure 2 and Table 4.

Answer group

a) Account activated and Ready to use

b) Activation Key sent

c) Activation Key resent

d) Close account

e) Logged out

f) Message sent

g) No user action

Q6. Read the following description of a program and the program itself, and then answer Subquestions 1 and 2.

[Program Description]

`PointInPolygon` is a subprogram that determines whether a given point is inside a given polygon. A polygon of `n`-vertices is represented as two arrays of size `n`, namely `poly_x` and `poly_y`, which corresponds to the *x* and *y* coordinates, respectively. The subprogram returns true if a given point lies inside the polygon.

Figure 1 illustrates the vertices and edges of a polygon.
Each pair of *x* and *y* coordinates represent a vertex on the polygon. For example, `poly_x[0]` and `poly_y[0]` correspond to the first vertex of a given polygon.
Additionally, two consecutive vertices represent an edge. For example, the pairs (`poly_x[0]`, `poly_y[0]`) and (`poly_x[1]`, `poly_y[1]`) corresponds to the first edge. The last edge is identified by the pairs (`poly_x[n-1]`, `poly_y[n-1]`) and (`poly_x[0]`, `poly_y[0]`).

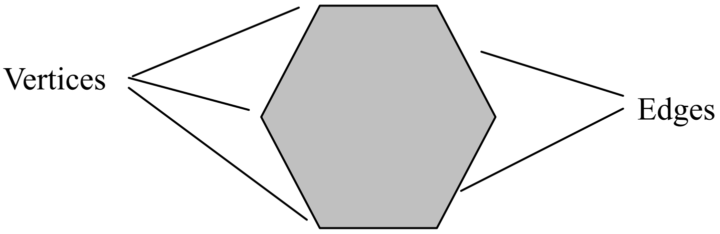


Figure 1. Vertices and edges of a 6-sided polygon

The subprogram `PointInPolygon` uses a simple ray-casting algorithm. The algorithm involves going through each of the edges of the polygon and testing if the ray projected from the given point intersects with the edge. If the total number of intersections is even, then the point is outside, otherwise, the point lies inside the polygon.

This approach can be visualized by projecting an imaginary ray originating from the point to the right side of the polygon as shown in Figure 2.

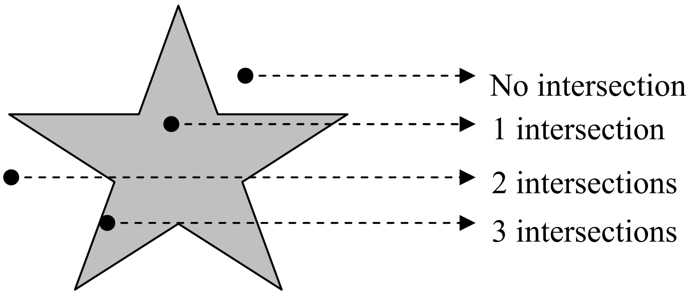


Figure 2. Ray-casting algorithm

`PointInPolygon` calls the `TestIntersect` subprogram to test for intersection. `TestIntersect` returns true if the ray projected to the right intersects the edge.

For testing intersections, the `TestIntersect` subprogram uses the following steps:

1. If the *y*-coordinate of the point is greater than the minimum *y* and less than or equal to the maximum *y* of the edge, proceed to step 2, otherwise proceed to step 6.
2. If the *x*-coordinate of the point is less than or equal to the maximum *x* of the edge, proceed to step 3, otherwise proceed to step 6.
3. If the edge is a vertical line, return true, otherwise proceed to step 4.
4. Compute for the *x*-coordinate of the intersection point using the following formula:

```
intersection_x = pt1_x + (pt_y - pt1_y) * (pt2_x - pt1_x) / (pt2_y - pt1_y)
```

5. If the *x*-coordinate of the point is less than or equal to the *x*-coordinate of the intersection point, return true, otherwise proceed to step 6.
6. Return false.

Assuming that a point to be tested does not lie on the boundary of the polygon, that is, a point does not lie on any edge including its vertices.

The functions `min` and `max` return the minimum and maximum values of the given parameters, respectively.

[Program]

```
○ logical: PointInPolygon(real:poly_x[], real:poly_y[],
                          integer:num_vertices, real:pt_x, real:pt_y)
o integer: intersect_count, index_1, index_2
o real: pt1_x, pt1_y, pt2_x, pt2_y
• index_1 ← 0
• intersect_count ← 0
■ index_1 < num_vertices
│ • pt1_x ← poly_x[index_1]
│ • pt1_y ← poly_y[index_1]
│ • index_2 ← [      A      ]
│ • pt2_x ← poly_x[index_2]
│ • pt2_y ← poly_y[index_2]
│ ▲ TestIntersect(pt1_x, pt1_y, pt2_x, pt2_y, pt_x, pt_y) = true
│ │ • intersect_count ← [      B      ]
│ ▼
│ • index_1 ← index_1 + 1
■
▲ [      C      ]
│ • return false
▼
• return true

○ logical: TestIntersect(real:pt1_x, real:pt1_y,
                         real:pt2_x, real:pt2_y,
                         real:pt_x,  real:pt_y )
○ real: intersection_x
▲ pt_y > min(pt1_y, pt2_y) and pt_y <= max(pt1_y, pt2_y)
│ ▲ pt_x <= max(pt1_x, pt2_x)
│ │ ▲ [      D      ]
│ │ │ • return true
│ │ ▼
│ │ • intersection_x ← pt1_x
│ │                 + (pt_y - pt1_y)*(pt2_x - pt1_x)/(pt2_y - pt1_y)
│ │ ▲ [      E      ]
│ │ │ • return true
│ │ ▼
│ ▼
▼
• return false
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank □ in the program.

Answer group for A

a) `(index_1 % num_vertices) + 1`
b) `(index_1 + 1) % num_vertices`
c) `min(index_1 + 1, 0)`
d) `max(index_1 + 1, num_vertices - 1)`

Answer group for B

a) `1`
b) `index_1`
c) `index_2`
d) `intersect_count + 1`

Answer group for C

a) `(intersect_count % 2) = 0`
b) `(intersect_count % 2) = 1`
c) `(intersect_count / 2) = 0`
d) `(intersect_count / 2) = 1`

Answer group for D

a) `pt1_x = pt2_x`
b) `pt1_x ≠ pt2_x`
c) `pt1_y = pt2_y`
d) `pt1_y ≠ pt2_y`

Answer group for E

a) `pt_x <= intersection_x`
b) `pt_x >= intersection_x`
c) `pt1_x <= intersection_x`
d) `pt1_x >= intersection_x`

Subquestion 2

Given an 8-sided polygon with the following vertices: `(0,0)`, `(4,0)`, `(4,4)`, `(8,4)`, `(8,0)`, `(12,0)`, `(12,8)`, `(0,8)`. Then, the program is executed to test if two points `(0,2)` and `(4,2)` are inside or outside the polygon.
From the answer group below, select the correct answer which indicates the execution results.

Answer group

a) Point `(0,2)` and point `(4,2)` are both inside the polygon
b) Point `(0,2)` and point `(4,2)` are both outside the polygon
c) Point `(0,2)` is inside while point `(4,2)` is outside the polygon
d) Point `(0,2)` is outside while point `(4,2)` is inside the polygon

Concerning questions **Q7** and **Q8**, **select one** of the two questions.
Then, mark Ⓢ in the selection area on the answer sheet, and answer the question.
If two questions are selected, only the first question will be graded.

Q7. Read the following description of a C program and the program itself, and then answer Subquestions 1 and 2.

[Program Description]

The function `convert` converts a specified path to an absolute path.

In a file system with a hierarchical structure, strings for identifying files and directories are called paths. Paths described with the root directory as the starting point are called absolute paths. They start with a "`/`" and each of the hierarchy is delimited by "`/`". When the specified path is a directory, the last "`/`" can be omitted. For example, an absolute path for directory `e` in the Figure can be represented as either "`/a/d/e`" or "`/a/d/e/`".

On the other hand, paths described with the current directory as the starting point are called relative paths. In a relative path, "`..`" is used to specify the level immediately above the current level. For example, when directory `c` in the Figure is the current directory, the relative path for file `file1.txt` can be represented as "`../file1.txt`", and the relative path for directory `e` can be represented as "`../../d/e`" or "`../../d/e/`". The current directory itself is represented as "`.`" or "`./`".

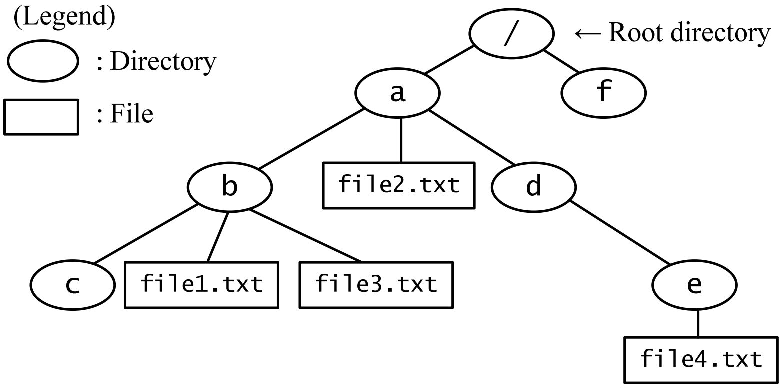


Figure Example of a file system with a hierarchical structure

(1) The specification of the function is as follows:

```
void convert(const char *path,
             const char *base,
             char *result);
```

Arguments: path – The path (before conversion)

base – The absolute path for the current directory

result – The absolute path (after conversion)

Function: If path is a relative path described based on base, it is converted to an absolute path, and stored into result. If path is an absolute path, the function copies path to result as is, irrespective of base.

Return value: None.

Here, the area referenced by result is large enough to store the converted string. In addition, redundant paths or strings that cannot be recognized as a path are not passed as arguments.

(2) The function does not check whether the specified directories or files actually exist in the file system.

(3) Examples of the conversion are shown in Table 1.

Table 1 Examples of Conversion

path	base	result
../../b/c/	/a/d/e/	/a/b/c/
b/file1.txt	/a/	/a/b/file1.txt
c/	/a/b/	/a/b/c/
file1.txt	/a/b/	/a/b/file1.txt
./	/a/b/c/	/a/b/c/
/a/d/e/file4.txt	/a/b/c/	/a/d/e/file4.txt

(4) This function uses the following library functions:

```
unsigned int strlen(const char *s);
```

Function: Computes the length of character string s.

Return value: Returns the number of characters preceding the null character that indicates the end of the string.

```
int strcmp(const char *s1, const char *s2);
```

Function: Compares character strings s1 and s2.

Return value: Returns 0 if s1 and s2 are identical, otherwise returns a non-zero.

```
int strncmp(const char *s1, const char *s2, int n);
```

Function: Compares character strings s1 and s2 up to the first n-th character (or up to the null character).

Return value: Returns 0 if the compared n characters are identical, otherwise returns a non-zero (including cases in which the comparison ends prematurely).

`char *strcpy(char *s1, const char *s2);`

Function: Copies character string `s2` (including the null character) to character string `s1`.

Return value: `s1`

`char *strncpy(char *s1, const char *s2, int n);`

Function: Copies `n` characters of character string `s2` to character string `s1`. If the length of `s2` is longer than or equal to `n`, the first `n` characters are copied. If the length of `s2` is shorter than `n`, the remainder is filled with null characters.

Return value: `s1`

[Program]

```
#include <string.h>

void convert(const char*, const char*, char*);

void convert(const char *path, const char *base, char *result){

   const char *pp, *bp;
   char *rp;
   int length;

   /* If path is an absolute path */
   if(*path == '/'){
      [          A          ];
      return;
   }
   /* If path indicates the current directory itself */
   if(!strcmp(path, ".") || !strcmp(path, "./")){
      [          B          ];
      return;
   }

   length = strlen(base);
   bp = base + length;     /* bp points to the end of character string base */
   if(*(bp - 1) == '/')
      --bp;

   /* By analyzing ".." or "../" at the beginning of path,
      determine the part of base that should be copied to result. */
```

```
    for(pp = path; *pp != '\0' && *pp == '.';){
       if(!strncmp(pp, "../", 3)){
          pp += 3;
          while(bp > base && *--bp != '/');
       }else if(!strncmp(pp, "./", 2)){
          pp += 2;
       }else if(!strncmp(pp, "..\0", 3)){
          pp += 2;
          while(bp > base && *--bp != '/');
       }else{
          break;
       }
    }
    /* Copy the part of base to result. */
    length =           C           ;
    strncpy(result, base, length);

    rp =           D           ;
    *rp++ = '/';

    /* Append the part of path excluding the leading "./" or "../"
       (substring pointed by pp) to result. */
    strcpy(rp, pp);
    return;
 }
```

Subquestion 1

From the answer groups below, select the correct answer to be inserted into each blank □ in the program.

Answer group for A and B

a) `strcpy(base, path)`	b) `strcpy(base, result)`
c) `strcpy(path, base)`	d) `strcpy(path, result)`
e) `strcpy(result, base)`	f) `strcpy(result, path)`

Answer group for C

a) `bp - base`	b) `bp - path`
c) `pp - base`	d) `pp - path`

Answer group for D

a) `base + length`	b) `bp + length`
c) `path + length`	d) `pp + length`
e) `result + length`	

Subquestion 2

Concerning the behavior of the program when the function `convert` is called with the arguments shown in Table 2, from the answer groups below, select the correct answer to be inserted into each blank □ in Table 2. Assuming that blanks A through D in the program contain the correct answers.

Table 2 Arguments

path	base	result
`../../../../d`	`/a/b/c/`	`/d/`
`d/`	`/a/b/c`	E
`d`	`/a/b/c/`	F

Answer group for E and F

a) `/a/b/c/d`	b) `/a/b/c/d/`
c) `/a/b/cd`	d) `/a/b/cd/`
e) `d`	f) `d/`

Q8. Read the following description of a Java program and the program itself, and then answer Subquestion.

[Program Description]

The program calculates and prints out the perimeter and the area of the geometric objects.
There are two types of geometric objects: a circle and a rectangle.
Each type of geometric objects is created by the corresponding constructor.
The program has the following functions:

(1) To print out objects' attributes and results of the calculation
(2) To draw objects (messages only)
(3) To print out the total size of areas

There is an interface `GraphicObjectTypes` that stores the type of the geometric objects.
The program consists of methods of a base class, its derived class, and `main()` method.
The class `Helper` provides all information for the output. Data is transferred by constructor function in class `Helper`.
Each class is stored in a separate file.

When executing this program, the following list will be printed out.

```
Circle Radius = 3.0
Size = 18.84955592153876 area = 28.274333882308138

Drawn circle
------------------------------
Rectangle width = 10.0 and length = 2.0
Size = 24.0 area = 20.0

Drawn rectangle
------------------------------
Circle Radius = 12.8
Size = 80.4247719318987 area = 514.7185403641517

Drawn circle
------------------------------
------------------------------
Total size of Area = 562.9928742464599
```

[Program]

```
public interface GraphicObjectTypes {
   final int GRAPHICOBJECT = 0;
   final int RECTANGLE = 1;
   final int CIRCLE = 2;
}

public class GraphicObject {
   private double size() {
      return 0;
   }
   private double area() {
      return 0;
   }
   public void draw() {
      System.out.println("Graphic Object drawed");
   }
   public       A       getType() {
      return GraphicObjectTypes.GRAPHICOBJECT;
   }
}

public class Circle extends GraphicObject
         B       GraphicObjectTypes {
   private double radius = 1.0;
   public Circle() {
   }
   public Circle(double r) {
      this.radius = r;
   }
   public double getRadius() {
      return radius;
   }
   public void setRadius(double r) {
      if (r > 0)
         this.radius = r;
   }
   public double size() {
      return 2 * Math.PI * radius;
   }
   public double area() {
      return Math.PI * radius * radius;
   }
   public void [      C      ] {
      System.out.println("Drawn circle");
   }
   public int getType() {
      return CIRCLE;
   }
   public String [      D      ] {
      return "Circle Radius = " + this.radius + "\n"
             + "Size = " + size() + " area = " + area() + "\n";
   }
}

public class Rectangle extends GraphicObject
       [      B      ] GraphicObjectTypes {
   private double width = 1.0;
   private double length = 1.0;
   public Rectangle() {
   }
   public Rectangle(double b, double l) {
      this.width = b;
      this.length = l;
   }
   public double getWidth() {
      return width;
   }
   public double getLength() {
      return length;
   }
   public void setWidth(double b) {
      if (b > 0)
         this.width = b;
   }
   public void setLength(double l) {
      if (l > 0)
         this.length = l;
   }
   public double size() {
      return 2 * (length + width);
   }
   public double area() {
      return length * width;
   }
   public void [      C      ] {
      System.out.println("Drawn rectangle");
   }
   public String [    D    ] {
      return "Rectangle width = " + this.width
             + " and length = " + this.length + "\n"
             + "Size = " + size() + " area = " + area() + "\n";
   }
   public int getType() {
      return RECTANGLE;
   }
}

public class Helper {
   GraphicObject[] graphicObjects;
   public Helper() {
      graphicObjects = new GraphicObject[3];
      graphicObjects[0] = new Circle(3);
      graphicObjects[1] = new Rectangle(10, 2);
      graphicObjects[2] = new Circle(12.8);
   }
   public double getTotalSizeOfArea() {
      double totalSize = 0.0;
      for (int i = 0; [    E    ]; ++i) {
         if (graphicObjects[i].getType()
             == GraphicObjectTypes.CIRCLE) {
            totalSize += [    F    ].area();
         } else {
            totalSize += [    G    ].area();
         }
      }
      return totalSize; // totalSize
   }
   public void printInfo() {
      for (int i = 0; [    E    ]; ++i) {
         if (graphicObjects[i].getType()
             == GraphicObjectTypes.CIRCLE) {
            System.out.println(graphicObjects[i]);
            graphicObjects[i].draw();
            System.out.println("------------------------------");
         } else {
            System.out.println(graphicObjects[i]);
            graphicObjects[i].draw();
            System.out.println("------------------------------");
         }
      }
   }
}
```

```
public class TestGObject {
   public static void main(String[] args) {
      Helper help = new Helper();
      help.printInfo();
      System.out.println("------------------------------");
      System.out.println("Total size of Area = "
                         + help.getTotalSizeOfArea());
   }
}
```

Subquestion

From the answer groups below, select the correct answer to be inserted into each blank □ in the above Program.

Answer group for A and B

- a) `final int`
- b) `GRAPHICOBJECT`
- c) `implements`
- d) `import`
- e) `int`
- f) `interface`

Answer group for C and D

- a) `draw`
- b) `draw()`
- c) `getDraw()`
- d) `getString()`
- e) `String`
- f) `string()`
- g) `this.draw()`
- h) `toString()`

Answer group for E

- a) `i < graphicObjects.length`
- b) `i < graphicObjects.length()`
- c) `i <= graphicObjects.length`
- d) `i <= graphicObjects.length()`

Answer group for F and G

- a) `((Circle) graphicObjects[i])`
- b) `((Rectangle) graphicObjects[i])`
- c) `(Circle graphicObjects[i])`
- d) `(Rectangle graphicObjects[i])`
- e) `Circle`
- f) `graphicObjects[i]`
- g) `GraphicObjectTypes.CIRCLE`
- h) `GraphicObjectTypes.RECTANGLE`