ANSWER SCHEME NEW – SET2
PSPM (CP015)

<table>
<tr><th>Question No.</th><th>Answer</th><th>Mark</th></tr>
<tr><td>1 (a)</td><td>1. Problem Analysis is the act of <u>identifying input, process and output</u>.

2. Design a solution involves <u>creating an algorithm</u> to solve the problem.

3. Implementation is <u>applying the algorithm</u> by writing a computer program using a programming.

4. Testing is the process to <u>ensure the program runs correctly</u> and is error free.

5. Documentation is a <u>written detailed description</u> of the programming cycle and <u>specific facts</u> about the program.</td><td>For each acceptable description
J1 X 5 = J5</td></tr>
<tr><td>1(b)(i)</td><td>
<table>
<tr><th>Decription</th><th>Input, Process or Output</th></tr>
<tr><td>Calculate average rating of e-hailing driver’s service in mobile application.

Menghitung kadar purata perkihmatan pemandu e-panggilan dalam aplikasi mudah alih.</td><td>Process</td></tr>
<tr><td>Produce receipt for an online transaction.
Menghasilkan resit bagi satu transaksi atas talian</td><td>Output</td></tr>
<tr><td>List all students enrolled in computer programming course.
Menyeranaraikan semua pelajar yang telah mendaftar dalam kursus pengaturcaraan komputer</td><td>Output</td></tr>
<tr><td>User activate live sharing location function on their smartphone.
Pengguna mengaktifkan fungsi lokasi perkongsian langsung pada telefon pintar.</td><td>Input</td></tr>
<tr><td>Password verification for email application.
Pengesahan kata laluan untuk aplikasi email.</td><td>Process</td></tr>
</table>
</td><td>J1 X 6 = J6</td></tr>
</table>

<table>
<tr><th>Question No.</th><th>Answer</th><th>Mark</th></tr>
<tr><td></td><td>
<table>
<tr><td>The usage of fingerprint for new passport application.
Penggunaan cap jari bagi permohonan passport baru.</td><td>Input</td></tr>
</table>
</td><td></td></tr>
<tr><td>1(b)(ii)</td><td>
<table>
<tr><td>Input
Input</td><td>1. Total purchase.
2. Membership status.</td></tr>
<tr><td>Process
Proses</td><td>1. Determine discount based on <u>total purchase</u> and <u>membership status.</u>
2. Calculate discounted price.</td></tr>
<tr><td>Output
Output</td><td>1. Total discount.
2. Discounted price.</td></tr>
</table>
</td><td>J1 X 2 for input

J2 for process 1.
J1 for process 2.

J1 X 2 for output

Total = J7</td></tr>
<tr><td>1(b)(iii)</td><td>
<table>
<tr><td>Input
Input</td><td>1. Total number of students.
2. Mark of the students.</td></tr>
<tr><td>Process
Proses</td><td>1. Repeatedly request for each student mark.
2. Calculate total mark.
3. Calculate average mark.</td></tr>
<tr><td>Output
Output</td><td>1. Total mark.
2. Average mark.</td></tr>
</table>
OR
<table>
<tr><td>Input
Input</td><td>1. Total number of students (n)
2. Mark of the n students.</td></tr>
<tr><td>Process
Proses</td><td>1. Repeatedly request for n student mark.
2. Calculate total mark.
3. Calculate average mark.</td></tr>
<tr><td>Output
Output</td><td>1. Total mark.
2. Average mark.</td></tr>
</table>
</td><td>J1 X 2 for input

J1 X 3 for process

J1 X 2 for output

Total = J7</td></tr>
</table>

<table>
<tr><th>Question No.</th><th>Answer</th><th>Mark</th></tr>
<tr><td>2(a)</td><td>Start
Read total duration(J1)
If duration<60 minutes(J1)
Chargers = RM5(J1)
Else if duration <120 minutes(J1)
Chargers = RM5 + RM0.20*(Duration-60) (J1)
Else(J1)
Chargers = RM5 + RM0.20*(Duration-120) (J1)
Display Charges(J1)
Stop</td><td>Total =J8

Missing Start/Stop: deduct J1</td></tr>
<tr><td>2(b)</td><td>An algorithm is a <u>step-by-step procedure/set of rules (J1)</u> designed to <u>solve a specific problem/accomplish a particular task (J1)</u></td><td>Total = J2</td></tr>
<tr><td>3(a)</td><td><table><tr><th></th><th>Procedural programming
Pengaturcaraan prosedural</th><th>Object-oriented programming
Pengaturcaraan berorientasikan objek</th></tr><tr><td>Definition
Takrifan</td><td>Procedural programming is a programming paradigm that focuses on describing a series of steps or procedures for the computer to execute to solve a problem or perform a task.</td><td>Object-oriented programming (OOP) is a programming paradigm that revolves around the concept of "objects," which are instances of classes. In OOP, objects encapsulate data (attributes) and behaviors (methods or functions) that operate on the data.</td></tr><tr><td>Example of programming language
Contoh bahasa pengatucaraan</td><td>C
FORTRAN
Pascal
Basic</td><td>C++
Python
JAVA
C#</td></tr></table></td><td>J1 X 2 =J2 for definitions

J1 X 2=J2 for examples

(any possible example accepted)

Total =J4</td></tr>
<tr><td>3(b)</td><td><table><tr><th>Identifier
Pengecam</th><th>Valid / Invalid
Sah / Tidak Sah</th></tr><tr><td>_myname_</td><td>Valid</td></tr><tr><td>my name</td><td>Invalid</td></tr><tr><td>@myname</td><td>Invalid</td></tr></table></td><td>J1 X 3 =J3</td></tr>
</table>

<table>
<tr><th>Question No.</th><th>Answer</th><th>Mark</th></tr>
<tr><td>3(c)</td><td>
<table>
<tr><th>Variable
Pemboleh ubah</th><th>Value
Nilai</th><th>Data Type
Jenis Data</th></tr>
<tr><td>IC_number</td><td>800511-10-4478</td><td>String</td></tr>
<tr><td>height</td><td>1.65</td><td>Float OR double</td></tr>
<tr><td>is_member</td><td>TRUE</td><td>Boolean OR Bool</td></tr>
</table>
</td><td>J1 X 3=J3</td></tr>
<tr><td>3(d)(i)</td><td>Volume = 4/3*(22/7)*(r*r*r)

OR

Volume = 4/3*(22/7)*(r**3)</td><td>J1 for (r*r*r)/(r**3)

J1 for others *accept answer without bracket

Total =J2</td></tr>
<tr><td>3(d)(ii)</td><td>a=(b+c)/(d-e)+f</td><td>J1 for (b+c)/(d-e)

J1 for others

Total =J2</td></tr>
<tr><td>3(e)</td><td>height=double(input(“Enter your height:”)</td><td>J1</td></tr>
<tr><td>4(a)</td><td>Sum: 12
Difference: 8
Product: 20
Division: 5.0
Modulus: 0</td><td>J1 X 5 =J5</td></tr>
<tr><td>4(b)</td><td>
<table>
<tr><th></th><th>Definition
Takrifan</th><th>Code Segment
Segmen Kod</th></tr>
<tr><td>Single Selection
Pilihan Tunggal</td><td>Selecting one path of execution based on the outcome of a single condition.</td><td>if <condition>:
  <statement></td></tr>
<tr><td>Dual Selection
Dwi Pilihan</td><td>Program execution to choose between two actions based on the value of the condition.</td><td>if <condition>:
  <statement>
else:
  <statement></td></tr>
</table>
</td><td>J1 X 2 = J2 for definition

J1 X 2 = J2 for example of code segment

Total =J4</td></tr>
</table>

Question No.	Answer	Mark
4(c)	`if monthly_income < 4800:` **(J1)** `    print("Status: B40")` **(J1)** `elif monthly_income < 7800:` **(J1)** `    print("Status: M40")` **(J1)** `else:` **(J1)** `    print("Status: T20")` **(J1)**	**Total =J6**
5(a)	`counter = int(input("Enter repetition: "))` **(J1)** `i = 0` **(J1)** `while i < counter:` **(J1)** `    print(i)` **(J1)** `    i += 1` **(J1)**	**Total =J5**
5(b)	`mark = int(input("Enter the student's mark (enter a negative value to finish): "))` **(J1)** `while mark >= 0:` **(J2)** `    print(mark)` **(J1)** `    mark = int(input("Enter the next student's mark (enter a negative value to finish): "))` **(J1)**	**Total =J5**
5(c)	`num_items = int(input("Enter the number of items purchased: "))` **(J0.5)** `total_price = 0` **(J1)** `for i in range(num_items):` **(J1)** `    price = float(input("Enter the price of item {i+1}: "))` **(J0.5)** `    total_price += price` **(J1)** `print("The total price of all items purchased is:", total_price)` **(J1)**	**Total =J5**
6	`sum_even = 0` **(J1)** `sum_odd = 0` **(J1)** `num = int(input("Enter a positive number (or a negative number to quit): "))` **(J1)**	

Question No.	Answer	Mark
	`while num>0:` **(J1)** `    if num % 2 == 0:` **(J1)** `        sum_even += num`**(J1)** `    else:` **(J1)** `        sum_odd += num`**(J1)** `    num = int(input("Enter a positive number (or a negative number to quit): "))`**(J1)** `print("Sum of even numbers:", sum_even)` **(J0.5)** `print("Sum of odd numbers:", sum_odd)` **(J0.5)**	**Total =J10**

END OF ANSWER SCHEME