Question	Part	Marking guidance	Total marks
1		**4 marks for AO3 (design), 7 marks for AO3 (program)** Any solution that does not map to the mark scheme refer to lead examiner **Note to Examiners:** For marks **E** and **J** be careful not to penalise the same error twice. For example, if they have used 6 instead of 7 in mark E and then 21 instead of 22 in mark J apply a **DPT** **Program Design** **Note** that AO3 (design) marks are for selecting appropriate techniques to use to solve the problem, so should be credited whether the syntax of programming language statements is correct or not and regardless of whether the solution works. **Mark A** for attempting to randomly generate **two** numbers; **Mark B** for use of selection to check the current score against 21; **Mark C** for using iteration to keep rolling the dice; **Mark D** for outputting the dice rolls in appropriate places; **Program Logic** **Mark E** for generating **two** random numbers between 1 and 6 inclusive; **Mark F** for correctly adding the **two** dice values cumulatively to the previous score; **Mark G** for a loop that terminates if the current score is less than 21 **and** player chooses not to roll again; **Mark H** for a correct mechanism to end the game if the player has a score greater than or equal to 21; **Mark I** for a selection statement which correctly checks if the player has lost (final score is greater than 21) **OR** won (final score is 21); **Mark J** for generating a random number between 15 and 21 inclusive in a logically correct place **AND** checking if the result is greater than the final score; **Mark K** for **at least one** correct set of messages output in appropriate places to show whether the user has won or lost; **A.** yes/y, no/n or any other appropriate equivalents **Maximum 10 marks** if any errors in code. **I.** Case **I.** Gaps/spaces throughout the code, except where to do so would explicitly alter the logic of the code in a way that makes it incorrect. **I.** Messages or no messages with input statements	11

C# Example 1 (fully correct)

All design marks are achieved (**Marks A**, **B**, **C** and **D**)

```
Random r = new Random();
int score = 0;
string rollAgain = "yes";

while (rollAgain == "yes")                                  (C, Part of G, Part of H)
{
   int dice1 = r.Next(1, 7);                                (Part of A,E)
   int dice2 = r.Next(1, 7);                                (Part of A,E)
   score = score + dice1 + dice2;                           (F)
   Console.WriteLine("Roll 1: " + dice1);                   (Part of D)
   Console.WriteLine("Roll 2: " + dice2);                   (Part of D)
   Console.WriteLine("Current score: " + score);            (Part of D)
   if (score < 21)                                          (Part of G)
   {
      rollAgain = Console.ReadLine();                       (Part of G)
   } else
   {
      rollAgain = "no";                                     (Part of H)
   }
}
if (score > 21)                                             (Part of I)
{
   Console.WriteLine("You lost! ");                         (Part of K)
} else if (score == 21)                                     (Part of I)
{
   Console.WriteLine("You won! ");                          (Part of K)
} else                                                      (Part of I)
{
   if (r.Next(15, 22) > score)                              (J)
   {
      Console.WriteLine("You lost! ");                      (Part of K)
   } else
   {
      Console.WriteLine("You won! ");                       (Part of K)
   }
}
```

I. Indentation in C#

A. `Write` in place of `WriteLine`

Python Example 1 (fully correct)

All design marks are achieved (**Marks A**, **B**, **C** and **D**)

```
import random
score = 0
rollAgain = "yes"

while rollAgain == "yes":                                   (C, Part of G, Part of H)
   dice1 = random.randrange(1, 7)                           (Part of A,E)
   dice2 = random.randrange(1, 7)                           (Part of A,E)
   score = score + dice1 + dice2                            (F)
   print(f"Roll 1: {dice1}")                                (D)
   print(f"Roll 2: {dice2}")
   print(f"Current score: {score}")
   if score < 21:                                           (Part of G)
      rollAgain = input()                                   (Part of G)
   else:
      rollAgain = "no"                                      (Part of H)
if score > 21:                                              (Part of I)
   print("You lost! ")                                      (Part of K)
elif score == 21:                                           (Part of I)
   print("You won! ")                                       (Part of K)
else:                                                       (Part of I)
   if random.randrange(15,22) > score:                      (J)
      print("You lost!")                                    (Part of K)
   else:
      print("You won! ")                                    (Part of K)
```

A. `random.randint(1, 6)`

A. `random.randint(15, 21)`

VB.NET Example 1 (fully correct)

All design marks are achieved (**Marks A**, **B**, **C** and **D**)

```
Dim r As Random = New Random()
Dim score As Integer
Dim rollAgain As String = "yes"
Dim dice1, dice2 As Integer

While rollAgain = "yes"                                          (C, Part of G, Part of H)

   dice1 = r.Next(1, 7)                                          (Part of A,E)
   dice2 = r.Next(1, 7)                                          (Part of A,E)
   score = score + dice1 + dice2                                 (F)
   Console.WriteLine("Roll 1: " & dice1)                         (Part of D)
   Console.WriteLine("Roll 2: " & dice2)                         (Part of D)
   Console.WriteLine("Current score: " & score)                  (Part of D)
   If score < 21 Then                                            (Part of G)
      rollAgain = Console.ReadLine()                             (Part of G)
   Else
      rollAgain = "no"                                           (Part of H)
   End If
End While

If score > 21 Then                                               (Part of I)
   Console.WriteLine("You lost! ")                               (Part of K)
ElseIf score = 21 Then                                           (Part of I)
   Console.WriteLine("You won! ")                                (Part of K)
Else                                                             (Part of I)
   If r.Next(15, 22) > score Then                                (J)
      Console.WriteLine("You lost! ")                            (Part of K)
   Else
      Console.WriteLine("You won! ")                             (Part of K)
   End If
End If
```

I. Indentation in VB.NET

A. `Write` in place of `WriteLine`

Question	Part	Marking guidance	Total marks

02	1	**2 marks for AO3 (refine)**
		
		Mark A for using the correct variable name and assigning a numeric value to it;
		Mark B for using correct code to generate a random number;
		
		Maximum 1 mark if any errors in code.
		
		I. Case
		I. Gaps/spaces throughout the code, except where to do so would explicitly alter the logic of the code in a way that makes it incorrect
		
		C#
		`randomNumber = rGen.Next(1, 101)  ;;`
		
		Python
		`randomNumber = random.randrange(1, 101)  ;;`
		
		VB.NET
		`randomNumber = rGen.Next(1, 101)  ;;`
			2