Pearson
BTEC

# BTEC
# Assignment Brief

## Unit 4
## Programming

## The story of the Task: Design a Tournament Scoring System

Imagine you are a new employee at a small software company. Your company recently gave a talk at a local college, and the college was very happy with it. Now, they have asked your company to help them run a tournament where students will compete in different events to win prizes.

Here's what you need to know about the tournament:

- **Participants:** Students can enter the tournament either alone (as an individual) or in teams.

There will be 4 teams with 5 members each.
There will be 20 individual competitors (students competing alone).

- **Events:** Each team or individual will compete in 5 different events.

Some events will be for teams, and some will be for individuals.
Events can include both sports (like running races) and academic challenges (like solving problems).

- **Scoring:** Points will be given based on how well each team or individual does in each event. The points system hasn't been decided yet, so the college wants to hear your ideas on how to score the events.

- **Flexible Participation:** Students can choose to compete in all 5 events or just one event if they want.

## Instructions for Task 1 what you need to do

- **Steps:**

1. **Design the Program:**

Think about the steps of creating software: First, you design the program; then you write the code, test it, and improve it.

Plan your system: What tasks does your program need to do? (Example: It needs to add points, track each student or team's score, and show who won.)

Create simple flowcharts or diagrams: Use pictures or charts to explain how your program will work. (Example: Draw a diagram showing how the program moves from one event to the next.)

Describe how the program will store the data: (Example: Store each student's name, team, and points in a list or database.)

2. **Programming Choices:**

Pick a programming language: Think about which programming language you'll use (like Python). Explain why you chose it.

Include pre-written code if possible: If you find any useful code that's already written (like for sorting lists), use it and explain how it helped.

3. **Test Your Program:**

Test plan: Write down what parts of the program you'll test. (Example: You'll check if it correctly adds up points.)

Test with real data: After you finish, run your program and write down any errors you find. Explain how you fixed them.

4. **Get Feedback:**

Ask for opinions from classmates or teachers about your program.

Improve your program based on their feedback and document the changes.

## Guidelines for this task


**Before you write any code, you need to plan how your program will work.**

**1. What to include in your design:**

The first stage is the planning stage. In this stage, you think about what the program needs to do.

Next comes design, where you figure out how the program will work (what tasks it will do, how it will store data).

**After that, you will develop the program (write the actual code). Finally, you will test your program to make sure it works.**

**Example: If you use Python, you might use simple loops to calculate scores or store data in lists.**

**2. Development**

Once you have planned the program, it's time to write the code and make the program work.

Things to focus on:

Efficiency: Use library functions that make your code faster and more organized.

Testing: After writing your program, test it. Try giving it some example scores and check if it calculates the points correctly.

Example of Testing: Create a test plan like:

Test if the program correctly calculates the total points. Test if the program handles errors (like if someone enters the wrong data).

### 3. Evaluation

After writing and testing your program, you need to evaluate your work.
Key Questions:
Does your program meet the tournament's needs?
Did you get feedback from someone else (maybe a teacher or classmate)? What changes did you make based on the feedback?

### 4. What You Will Submit

A. Design Documents: Include all diagrams, code, and explanations.
B. Program Code: The code for your program.
C. Test Plans: What you tested and the results.
D. Evaluation Report: A summary of how well your program works and what could be improved.

5. **Report Presentation:** Make sure your report looks professional:

- Use Arial or Calibri font in size 12.
- Set your line spacing to 1.5 for easy reading.
- Include a cover page with your name, school name, task number, and the date.
- Use page numbers to make the report easier to navigate.

Keep your report well-structured so that it's easy to follow.

6. **File Submission:** Submit your work as a rar folder. Based on point 4 what you will submit.

Name your file like this:

YourName_SchoolName_U4Task2.rar.

**Deadline for submission**

**30/10/2024 6:00 PM**


**Do Not Copy from the Internet or Use AI Tools: Make sure you do your own research and write the report in <u>your own words</u>.**
**Copying from online sources or relying too much on AI tools can affect your grades and learning.**

## Assessment Criteria

### PASS

1. You completed a working program that solves the problem, but there might be a few small issues.
2. You showed basic understanding of the software development life cycle and used some feedback to improve your design.

### MERIT

1. Your program is well-structured, efficient, and fully meets the requirements.
2. You used feedback to make your design better and explained why you made certain decisions.

## DISTINCTION (EXCELLENT)

1. **You provided a detailed evaluation of the entire process and showed strong planning and coding skills.**
2. **Your program is not only functional but also optimized, and you justified every design and coding decision.**

You cannot achieve a Merit grade unless you have first achieved the Pass grade.

You cannot achieve a Distinction grade unless you have first achieved both Merit and Pass grades.