Module 16 — ES6 Practice Tasks

Scope Detective

Function Name Must be: describeDeclaration

একটি কোডিং একাডেমি নতুন শিক্ষার্থীদের var, let, const এর পার্থক্য বোঝাতে একটি ছোট Quiz Tool বানাচ্ছে। Tool-টি একটি keyword ইনপুট নেবে এবং সেই keyword দিয়ে Redeclare ও Reassign করা যায় কিনা তা বর্ণনা করবে।

Input

Function একটি Parameter গ্রহণ করবে — keyword (String): "var" / "let" / "const"

নিয়ম (Rules)

- "var" হলে Return করবে "Can redeclare, can reassign"
- "let" হলে Return করবে "Cannot redeclare, can reassign"
- "const" হলে Return করবে "Cannot redeclare, cannot reassign"

Validation

Return "Invalid" যদি keyword এই তিনটির একটিও না হয়।

টেস্ট কেস

Input	Output
"let"	Cannot redeclare, can reassign
"const"	Cannot redeclare, cannot reassign
"var"	Can redeclare, can reassign
"int"	Invalid

Movie Ticket Booking

Function Name Must be: bookTicket

একটি সিনেমা Booking App-এ Default ভাবে প্রতি Booking-এ 1টি Seat এবং Ticket প্রতি ৩00 টাকা ধরা হয়, যদি ইউজার আলাদা কিছু না দেয়। তোমার কাজ হলো ES6 Default Parameter দিয়ে এই System টি তৈরি করা।

Input

Function তিনটি Parameter গ্রহণ করবে — movie (String), seats (Number, default 1), pricePerSeat (Number, default 300)

Output / নিয়ম

- Total ক্যালকুলেট করতে হবে seats × pricePerSeat
- Return করতে হবে: "<movie>: <seats> seat(s), Total ৳<total>"

Validation

Return "Invalid" যদি movie String না হয় অথবা seats/pricePerSeat Negative Number হয়।

টেস্ট কেস

Input	Output
bookTicket("Dune")	Dune: 1 seat(s), Total ৳300
bookTicket("Dune", 3)	Dune: 3 seat(s), Total ৳900
bookTicket("Dune", 2, 450)	Dune: 2 seat(s), Total ৳900
bookTicket(123, 2)	Invalid

Receipt Generator

Function Name Must be: generateReceipt

একটি Grocery Shop-এর POS System প্রতিটি Purchase-এর পর একটি Multi-line Receipt প্রিন্ট করে। Template Literal ব্যবহার করে এই Receipt তৈরি করতে হবে।

Input

Function তিনটি Parameter গ্রহণ করবে — customerName (String), items (Array of String), total (Number)

Output / নিয়ম

- Template Literal (backtick) ব্যবহার করে Multi-line String Return করতে হবে

- Format exactly:Receipt for <customerName>Items: <item1, item2, ...>Total: ৳<total>

Validation

Return "Invalid" যদি items Array না হয় অথবা Empty হয়।

টেস্ট কেস

Input	Output
"Rakib", ["Pen","Book"], 150	Receipt for Rakib\nItems: Pen, Book\nTotal: ৳150
"Sadia", ["Milk"], 60	Receipt for Sadia\nItems: Milk\nTotal: ৳60
"Tanvir", [], 0	Invalid

BMI Calculator (Arrow Refactor)

Function Name Must be: calculateBMI

একটি Fitness App-এর পুরানো ES5 Function কে Arrow Function এ Refactor করতে হবে। Function টি Weight (kg) এবং Height (m) নিয়ে BMI বের করবে।

Input

Arrow Function দুইটি Parameter গ্রহণ করবে — weight (Number, kg), height (Number, m)

Output / নিয়ম

- BMI = weight / (height × height)
- Return করতে হবে BMI একটি Number হিসেবে, 2 Decimal Place পর্যন্ত (toFixed(2), তারপর Number এ Convert)

Validation

Return "Invalid" যদি weight বা height শূন্য বা Negative হয়।

টেস্ট কেস

Input	Output
70, 1.75	22.86

50, 1.6	19.53
60, -1.7	Invalid

Inventory Merge & Max Score

Function Name Must be: mergeInventory / highestScore

একটি দোকানের Two Branch-এর Stock Array Spread Operator দিয়ে Merge করতে হবে, এবং একটি Student-দের Score Array থেকে Spread দিয়ে Highest Score বের করতে হবে।

Input

mergeInventory(arr1, arr2) — দুইটি Array; highestScore(scores) — একটি Number Array

Output / নিয়ম

- mergeInventory: Spread Operator (...arr1, ...arr2) দিয়ে একটি Merged Array Return করবে
- highestScore: Math.max(...scores) দিয়ে Highest Number Return করবে

Validation

Return "Invalid" যদি Parameter Array না হয়।

টেস্ট কেস

Input	Output
mergeInventory([1,2],[3,4])	[1, 2, 3, 4]
highestScore([3,7,2,9,4])	9
highestScore("3,7,2")	Invalid

User Profile Extractor

Function Name Must be: extractUserInfo

একটি Social App-এর API একটি Nested User Object Return করে। Destructuring ব্যবহার করে Nested Data থেকে Name, Age এবং Hobby List-এর First Item বের করতে হবে।

Input

extractUserInfo(userObj) — userObj = { user: { name, age }, hobbies: [...] }

Output / নিয়ম

- Object এবং Array Destructuring ব্যবহার করতে হবে (Default এবং Rename সহ)
- Return করতে হবে: "<name> (<age>) likes <firstHobby>"
- hobbies Array Empty হলে firstHobby এর Default হবে "nothing yet"

Validation

Return "Invalid" যদি user.name না থাকে।

টেস্ট কেস

Input	Output
{user:{name:"Sadia",age:22}, hobbies:["reading","coding"]}	Sadia (22) likes reading
{user:{name:"Rafi",age:19}, hobbies:[]}	Rafi (19) likes nothing yet
{user:{age:30}, hobbies:[]}	Invalid

Inventory Object Manager

Function Name Must be: manageInventoryObject

একটি Warehouse System Object-ভিত্তিক Product Stock Manage করে। Action Parameter অনুযায়ী keys/values/entries বের করে, Property Delete করে, অথবা Object Seal/Freeze করে।

Input

manageInventoryObject(obj, action) — action হতে পারে: "keys" | "values" | "entries" | "delete:<propName>" | "seal" | "freeze"

নিয়ম (Rules)

- "keys" → Object.keys(obj) Return করবে
- "values" → Object.values(obj) Return করবে

- "entries" → Object.entries(obj) Return করবে
- "delete:propName" → propName Delete করে নতুন Object Return করবে
- "seal" → Object.seal(obj) করে Return করবে (নতুন Property Add করা যাবে না)
- "freeze" → Object.freeze(obj) করে Return করবে (কোনো Value পরিবর্তন করা যাবে না)

Validation

Return "Invalid" যদি obj Plain Object না হয় অথবা action এই List-এ না থাকে।

টেস্ট কেস

Input	Output
{a:1,b:2,c:3}, "keys"	['a', 'b', 'c']
{a:1,b:2,c:3}, "entries"	[['a',1],['b',2],['c',3]]
{a:1,b:2,c:3}, "delete:c"	{a:1, b:2}
{a:1}, "shrink"	Invalid

Object Looper

Function Name Must be: printObjectDetails

একটি Product Catalog System একই Object তিন ভাবে Loop করে দেখাতে চায় — for…in, for…of + Object.entries, এবং Object.entries + Array Destructuring।

Input

printObjectDetails(obj, loopType) — loopType: "forin" | "forofentries" | "entriesDestructure"

নিয়ম (Rules)

- "forin" → for…in দিয়ে প্রতি Key Loop করে "<key>: <value>" Array Return করবে
- "forofentries" → for…of (Object.entries(obj)) দিয়ে একই Format Return করবে
- "entriesDestructure" → Object.entries(obj).map এ [key, value] Destructure করে একই Format Return করবে

Validation

Return "Invalid" যদি loopType অচেনা হয়।

টেস্ট কেস

Input	Output
{fruit:"Mango",price:50}, "forin"	['fruit: Mango', 'price: 50']
{fruit:"Mango",price:50}, "entriesDestructure"	['fruit: Mango', 'price: 50']
{}, "loopThrough"	Invalid

ES6 Refactor Challenge

Function Name Must be: refactorToES6 (concept task)

নিচের ES5-Style Codeটি পুরোনো একটি Shopping Cart Total Calculator। তোমার কাজ হলো এটিকে পুরো পুরো ES6 দিয়ে Refactor করা (var → let/const, String Concat → Template Literal, Regular Function → Arrow Function, Loop → Spread/reduce), তারপর ChatGPT কে দিয়ে তোমার Refactor Verify করা।

Input

নিচের ES5 Snippetটি ইনপুট হিসেবে ধরো

Output / নিয়ম

- var সব দুর করে let / const দিয়ে বদলানো
- String Concatenation ('+') দুর করে Template Literal (`...`) দিয়ে বদলানো
- Regular Function → Arrow Function তে Refactor করো
- for-loop এর Total Sum একটি Array.reduce() অথবা Spread দিয়ে লিখো

Validation

Refactorএর পর মূল Function-এর Output অপরিবর্তিত থাকতে হবে।

স্টার্টার কোড

```
// ES5 — refactor this fully to ES6:
var calculateTotal = function(items) {
  var total = 0;
  for (var i = 0; i < items.length; i++) {
    total = total + items[i].price;
  }
```

```
  return 'Total: ' + total + ' Taka';
};
```

টেস্ট কেস

Input	Output
[{price:100},{price:250}]	Total: 350 Taka
[{price:80}]	Total: 80 Taka